Security Management System for 4G Heterogeneous Networks

PhD Thesis

Hani Ali Alquhayz

This thesis is submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Software Technology Research Laboratory

Faculty of Technology

De Montfort University

May 2015

DEDICATION

To my beloved parents

I dedicate this thesis to my beloved father, **Mr. Ali Alquhayz,** who has been a great source of inspiration, motivation, and support throughout my life.

I also dedicate this thesis to my beloved mother, **Mrs. Dalal**, who gave me love and care throughout my life, for everything that she sacrificed for me in her life. Without her love, care, prayers, and support, I would not have achieved my goals.

To my beloved family

I would like to dedicate this thesis to my beloved wife, **Mrs. Hind,** and to my beloved son Malik for the great moments they have given me in our life together. I thank them for their support and for the help they have given me in so many ways in this life.

To my beloved brothers and sisters

I also dedicate this thesis to my brothers and sisters for their inspiration, love, prayers and support.

Abstract

There is constant demand for the development of mobile networks to meet the service requirements of users, and their development is a significant topic of research. The current fourth generation (4G) of mobile networks are expected to provide high speed connections anywhere at any time. Various existing 4G architectures such as LTE and WiMax support only wireless technologies, while an alternative architecture, Y-Comm, has been proposed to combine both existing wired and wireless networks. Y-Comm seeks to meet the main service requirements of 4G by converging the existing networks, so that the user can get better service anywhere and at any time.

One of the major characteristics of Y-Comm is heterogeneity, which means that networks with different topologies work together to provide seamless communication to the end user. However, this heterogeneity leads to technical issues which may compromise quality of service, vertical handover and security. Due to the convergence characteristic of Y-Comm, security is considered more significant than in the existing LTE and WiMax networks. These security concerns have motivated this research study to propose a novel security management system. The research aims to meet the security requirements of 4G mobile networks, e.g. preventing end user devices from being used as attack tools. This requirement has not been met clearly in previous studies of Y-Comm, but this study proposes a security management system which does this.

This research follows the ITU-T recommendation M.3400 dealing with security violations within Y-Comm networks. It proposes a policy-based security management system to deal with events that trigger actions in the system and uses Ponder2 to implement it. The proposed system, located in the top layer of the Y-Comm

architecture, interacts with components of Y-Comm to enforce the appropriate policies. Its four main components are the Intelligent Agent, the Security Engine, the Security Policies Database and the Security Administrator. These are represented in this research as managed objects to meet design considerations such as extensibility and modifiability.

This research demonstrates that the proposed system meets the security requirements of the Y-Comm environment. Its deployment is possible with managed objects built with Ponder2 for all of the components of Y-Comm, which means that the security management system is able to prevent end user devices from being used as attack tools. It can also achieve other security goals of Y-Comm networks.

DECLARATION

I declare that the work described in my thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the Software Technology Research Laboratory (STRL), De Montfort University, United Kingdom. No part of the material described in this thesis has been submitted for the award of any other degree or qualification in this or any other university or college of advanced education.

I also declare that part of this thesis has been published in my following publications.

Publications

- H. Alquhayz, A. Al-Bayatti and A. Platt. "Security management system for 4G heterogeneous networks." In *Proceedings of the World Congress on Engineering*, vol. 2. London, UK. 2012. **Certificate of Merit (Student).**

- H. Alquhayz, A. Al-Bayatti and A. Platt. "Protecting the End User Device in 4G Heterogeneous Networks." In *IAENG Transactions on Engineering Technologies*, pp. 339-348. Springer Netherlands, 2013. **Book Chapter, Invited.**

Acknowledgement

First of All, I would thank Allah (God) for all his blessings, gifts, everything in my life, and this opportunity to do a PhD. I thank Allah for supporting me in facing the challenges in this life.

I would like to thank my first supervisor, Dr. Ali Al-Bayatti, for his support, motivation, patience, and guidance throughout the PhD period. I would thank him for everything he taught me. I learned a lot from Dr. Ali, and his support and understanding will never be forgotten. I ask Allah to reward him in this life and in the life hereafter. His feedback and comments have been a great source of inspiration for me. I really appreciate his great supervision and direction.

I would also thank my previous second supervisor, Dr. Amelia Platt, for her inspiration and positive kind words. I want to say thank you for her support and guidance.

My greatest thanks to my father and mother; their help, support and prayers have motivated me to achieve this thesis, and allowed me to achieve previous achievements in my life. I ask Allah to save them in this life and reward them in paradise in the life hereafter. I also thank my brothers and sisters for the many great things they have done for me.

Also, my special thanks to my wife for her help, support and company in this country. Her motivational words helped me to get through the difficult times and to write this thesis. I love you Hind, and wish a wonderful life for us. Also, I would thank my son

who gave me a smile and a warm welcome every day. Playing with him was a recharger for me.

I also wish to thanks all of the STRL staff and my colleagues. Studying at the STRL has been a great experience in my life. I would thank all the great friends that I made in STRL, especially Tareq Binjammaz and Yazeed Alsaway. Also, I would thank all my friends and neighbours in Leicester. I met a lot of friends here in Leicester, and our friendship will last for ever.

Thank you.

Table of Contents

List of Figures

List of Abbreviations

1G	The first generation of mobile networks
2G	The second generation of mobile networks
3G	The third generation of mobile networks
4G	The fourth generation of mobile networks
AAAC	Authentication, Authorisation, Accounting, and Cost
AEL	Applications Environments Layer
AKA	Authentication and Key Agreement
AMPS	Analogue Mobile Phone Systems
AR	Access Router
CA3C	Central Authentication, Authorisation, Accounting, and Cost
CDMA	Code Division Multiple Access
CQoSB	Central Quality of Service Broker
CTS	Core Transport System Layer
DA3C	Domain Authentication, Authorisation, Accounting, and Cost
DSSS	Direct Sequence Spread Spectrum
ECA	Event Condition Action

ETL	End Transport Layer
EUD	End User Device
Gbit/s	Giga bit per second
GSM	Global Systems for Mobile communications
HPL	Hardware Platform Layer
IA	Intelligent Agent
ITU	International Telecommunication Union
Kbit/s	Kilobit per second
LaSCO	Language for Security Constraint on Objects
LTE	Long Term Evolution
MAC	Media Access control Address
MAN	Metropolitan Area Network
MEID	Mobile Equipment Identifier
MHz	Mega Hertz
MIWSMS	Mobile IP-based Wireless Security Management System
NAL	Network Abstraction Layer
NAS	Network Architecture Security
NLA	Network-Level Agreement
NML	Network Management Layer

NQL	Network QoS Layer
NTS	Network Transport Security
OSI	Open Systems Interconnection model
PAF	Ponder2 Authorisation Framework
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PML	Policy Management Layer
PSTN	Public Switched Telephone Network
PWLSMS	Policy-based Wireless Local area network Security Management System
QBS	QoS-Based Security
QL	Quality of Service Layer
QoS	Quality of Service
REL	Reconfiguration Layer
RMI	Remote Method Invocation
SAS	Service and Application Security
SCI	System Call Interface
SLA	Service-Level Agreements
SMC	Self-Managed Cell
SMS4HN	Security Management System for 4G Heterogeneous Networks

SPL	Service Platform Layer
STRBAC	Spatio-temporal Role-based Access Control model
TACS	Total Access Communication Systems
TCP/IP	Transmission Control Protocol
TMN	Telecommunications Management Network
ToS	Theft of Service
UMTS	Universal Mobile Telecommunications System
VFO	Virgin Fibre Optic
VHL	Vertical Handover Layer
WiMax	Worldwide Interoperability for Microwave Access
XACML	Extensible Access Control Mark-up Language

Chapter 1: **Introduction**

Objectives

- Introduce the research and explain its motivation;
- Present the research questions;
- Present the research methodology;
- Present the thesis structure.

1.1 Introduction

Security management systems offer promising solutions for protecting heterogeneous networks. Part of their function is to reduce the human input required to manage a network, but they are driven by security needs and must respond instantly to malicious activities that threaten the network.

The Y-Comm framework, a significant architecture for 4G mobile networks, tends to meet users' service requirements for high speed and availability anywhere at any time. The heterogeneousness and openness of Y-Comm networks has created concerns about their security, which must be assured and verified. These concerns have been investigated, and it has been concluded that Y-Comm network security should be improved. This research study proposes a security management system that achieves the security requirements of 4G mobile networks. The proposed Security Management System for 4G Heterogeneous Networks (SMS4HN) is designed as a policy-based system.

This PhD research thesis offers a detailed account of the development and evaluation of the SMS4HN. Section 1.2 of this introductory chapter presents the motivation for the study, then Section 1.3 explains the scope of the thesis and Section 1.4 presents the research questions. The success criteria for the study are explained in Section 1.5, then Section 1.6 sets out the research methodology and Section 1.7 states the research hypotheses. Section 1.8 identifies the original contributions made by this PhD research, Section 1.9 explains the thesis structure and the chapter ends with an account of the design considerations.

1.2 Motivation

Many research studies have investigated ways to improve communication services for mobile network users. One of the most challenging problems in mobile computing, and in 4G heterogeneous networks in particular, is security [1, 2]. Several research studies have proposed Y-Comm for 4G heterogeneous networks. An important goal of Y-Comm networks is to provide reliable and seamless service everywhere at all times. Y-Comm is an open network, which means that the infrastructure is owned by different network providers. The open nature of Y-Comm leads to security risks and vulnerabilities, because if a single provider is compromised then the entire network will be affected [3].

Park and Park investigated the security of 4G heterogeneous networks and found that they suffer from several security threats, because such networks are IP-based and open [3]. The X.805 systematic tool was used in that study to investigate the security of 4G networks. In another study, Mahdi et al. [4] found that 4G heterogeneous networks suffer from more security threats than the previous (3G) generation of mobile networks, because each 3G network is owned by a single provider.

The security concerns associated with 4G heterogeneous networks motivated the present researcher to propose a solution that provides a secure environment. This research study proposes a policy-based security management system because these systems are known for their dynamic and efficient solutions. However, the nature of the Y-Comm network makes the implementation of a security management system difficult. This research

investigates several approaches and concludes that the Y-Comm network needs a novel security management system.

The security requirements of 4G heterogeneous networks have been identified by several research studies [5-7]. In the Y-Comm security model, one requirement which has not been clearly met is protecting end user devices (EUDs) from being used as attack tools. This also motivated this research study to develop the proposed security management system and meet the requirements of the 4G heterogeneous network.

1.3 Scope of thesis

Security management systems focus on four main tasks: detection, prevention, containment and recovery, and security administration. This research presents a mechanism that detects malicious events in EUDs, but the detection implementation is outside the scope of this thesis, which presents the proposed framework's prototype implementation for the security engine. This is the most important part of the system. This research also describes the main interaction with the Y-Comm network components. The implementation of the security engine demonstrates that the security management system is able to enforce policies in this heterogeneous environment. This research assumes the Y-Comm network components are able to provide an interface to interact with the managed objects and receive PonderTalk messages. The primary purpose of the system is to control the behaviour of the policy-based system to interact with the Y-Comm network components. This research shows that the security management system is able to respond to malicious events through its demonstrated

ability to respond to the theft of a user's identity, which may lead to further attacks on the Y-Comm network.

1.4 Research questions

The primary research question is:

Can Y-Comm security be improved by adopting a security management model?

To answer this question, this research investigated the security requirements of 4G mobile networks. One requirement was not clearly met, so it led to the following question:

How can an end user device in the Y-Comm network environment be prevented from being used as an attack tool?

To answer this question, this research requires that the proposed SMS4HN follow ITU-T Recommendation M.3400 when responding to malicious events.

The third question is:

Why does this research consider access to user's information more harmful in Y-Comm networks?

To answer this, the research explains the danger of accessing the user's identity and how it leads to attacks on the peripheral network and then the whole network.

1.5 Success criteria

The success of the proposed system is based on its ability to work in the Y-Comm network environment. The framework and prototype implementation seek to demonstrate that this environment is more secure when our proposed system is implemented. This research considers that the security management system should meet certain defined criteria to work efficiently in the Y-Comm heterogeneous network environment. These criteria include:

- The extensibility of the proposed system to respond to additional malicious events and to enforce policies without additional hardware;
- The ability to modify the behaviour of the proposed system without needing to suspend the system's services;
- The ability of the proposed policy-based system to be self-managed;
- Interoperability with the Y-Comm network components to enforce the policies.

1.6 Research methodology

The methodology used in this study was constructive research, because this suits research in computer science. Constructive research requires the new research to be developed via a framework, theory, model, or algorithm [8]. This research project developed the contribution in six stages: the first addressed the research background, the next four were scientific research work stages, and the final stage was the writing of this thesis. The details of these stages are as follows:

Stage 1: Research background

The background stage began with an overview of the research environment and the main concepts related to this research. This study used several resources, including the Google Scholar search engine, the ACM digital library, the British Library, and IEEE Xplore.

Stage 2: Framework

This stage focused on designing the SMS4HN framework to achieve the research objectives. It involved defining the components of the framework in detail, and explaining how these components interact with the Y-Comm network components.

Stage 3: Prototype implementation of the SMS4HN

This stage of the research explored the actions of the proposed system during a malicious event. It involved explaining the specific details of implementation and demonstrating the proposed system's ability to work in the Y-Comm network environment.

Stage 4: Case study

This next stage was to study how the SMS4HN would work in a real example of the Y-Comm network and to explain how the EUD would move from one peripheral network to another. Finally, this stage considered the intelligent agent (IA) detection mechanism, which is a component of the framework.

Stage 5: Evaluation of the proposed system

At the fifth stage, the proposed system was evaluated and compared to related research based on a set of criteria. This stage served to demonstrate the practical applicability of the proposed system to the Y-Comm network environment.

Stage 6: Write up

The final stage was to write up the work done in the earlier stages, in the form of a doctoral thesis.

1.7 Research hypotheses

Hypotheses are significant in the research process because they can be used as a guide for the preparation and development of the research. They also help to determine what data it is necessary to collect and analyse. "A hypothesis is a logical supposition, a reasonable guess, an educated conjecture. It provides a tentative explanation of a phenomenon under investigation" [9].

This section lists the hypotheses tested in this research, then Figure 1-1 illustrates their relations with the Y-Comm architecture and the proposed SMS4HN. The hypotheses are as follows:

H1- Stealing a user's identity leads to an attack on a peripheral network and then on the whole Y-Comm network.

H2- The different architecture of Y-Comm leads to more security concerns.

H3- This research follows ITU-T recommendations by providing better security management systems.

H4- The CA3C server in Y-Comm is able to interact with the proposed SMS4HN in order to enforce negative authorisation policies.

H5- The access router in Y-Comm is able to interact with the proposed SMS4HN in order to enforce negative authorisation policies.

H6- The IA is scalable and will work with various EUDs.

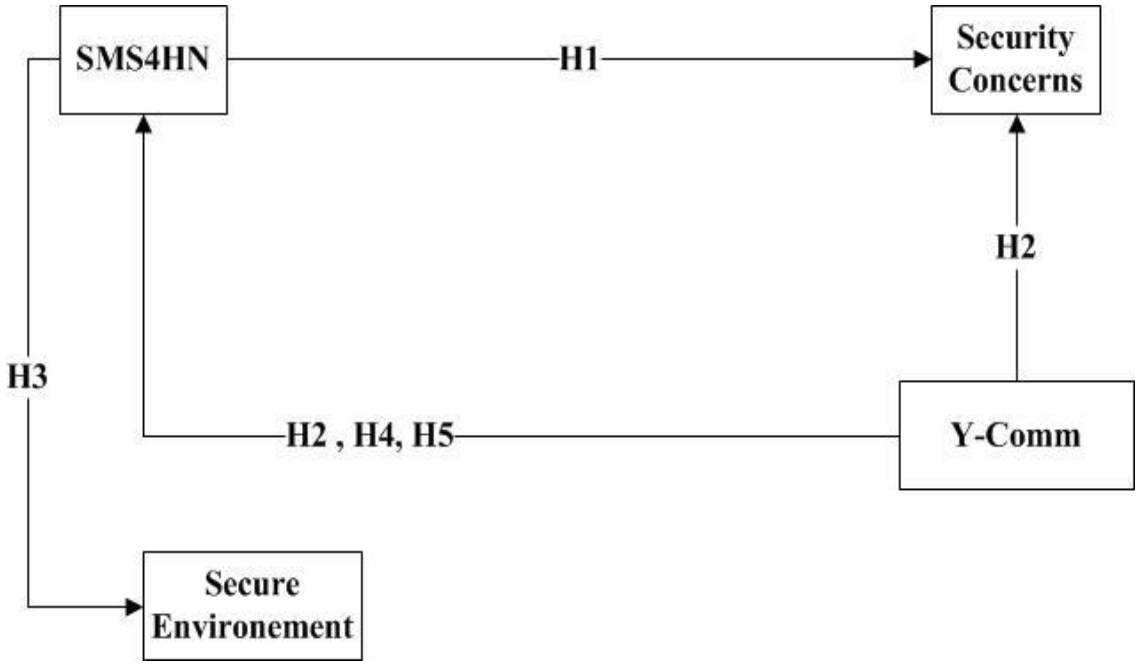

Figure 1-1. Relations between the hypotheses, Y-Comm, and SMS4HN

1.8 Original contribution

The main contribution of this research study is to propose a security management system suitable for the nature of the Y-Comm network. 'Suitable' here means that the system is able to enforce security policies in this complex architecture. The SMS4HN as presented is a novel security management system for Y-Comm. The different architecture of the Y-Comm network requires a novel security management system. This research study reviews the security issues affecting Y-Comm networks and the existing security management systems for similar wired and wireless networks, and addresses the gap in meeting the security requirements of 4G mobile networks. A policy-based system was chosen to achieve the research goals: to send messages to and control the behaviour of the components of Y-Comm networks.

The original contributions of this research study are as follows:

- A new approach to protecting Y-Comm networks.

  The approach presented prevents an EUD from being used as an attack tool in the Y-Comm environment. It follows ITU-T recommendation M.3400 to deal with the security violations in the network.

- A policy-based framework to enforce policies in Y-Comm networks.

  This framework is built on the top layer of the Y-Comm network model. Its purpose is to respond to malicious events that may harm the whole Y-Comm network. This research considers using extensible, modifiable, self-managed and

scalable components that will be able to interoperate with the heterogeneous components of the Y-Comm network.

- A self-managed cell to interact with the managed objects.

  This self-managed cell shows the interactions between the components of the SMS4HN and the Y-Comm network. The cell is triggered by the EUD and moves to the security engine, then ends by enforcing policies on the managed objects, such as the access router and the core endpoint server. Thus, it represents a policy feedback loop.

- An IA detection mechanism that detects malicious behaviour in an EUD.

  This IA mechanism explains how to detect malicious behaviour in an EUD that may result in compromised privileges or the theft of sensitive data. The mechanism as presented is limited to design descriptions and theories. This research specifies the appropriate mechanism for a specific smartphone operating system (Android OS).

  The novelty of this mechanism is that the IA in the EUD interacts with the security engine in the CAS3C, thus providing better security for the Y-Comm environment. Stealing users' identities is more harmful in this environment, because of the increase in user privileges on the network. Thus, the IA detects this kind of malicious event and reports it to the security engine.

1.9 Thesis structure

This section outlines the contents of each of the remaining chapters of this thesis.

Chapter 2: Background and Literature Review

Chapter 2 provides an overview of mobile computing, discussing the associated challenges and security issues. It then presents an overview of a 4G mobile network, specifically a Y-Comm network. Next, it gives details of the Y-Comm network architecture and security model, explaining the security concerns affecting Y-Comm networks and how important it is to address these. The fundamental concepts of policy are then explained, along with the motivation for the construction and use of a policy-based system in this research. There is then an account of the security policies adopted, more specifically the authorisation and obligation policies, followed by a summary of access control models. The chapter defines security management, discusses policy specification languages and justifies the use of Ponder2. Different approaches to security management systems are then discussed, with particular consideration of those which have been proposed to work on wireless and wired networks. Finally, the chapter explains why this research follows the ITU-T recommendations on responding to malicious events in the Y-Comm network.

Chapter 3: Preliminaries

Chapter 3 begins by discussing the structure of future heterogeneous networks, including Y-Comm networks. It then provides a justification for considering the access and theft of an end user's identity to be dangerous in this environment. Next, it

discusses the security requirements of 4G mobile networks and how some requirements have not been clearly met in the Y-Comm framework. It lists the assumptions made in this research study, then outlines the fundamentals and principles of Ponder2, such as the policy authorisation framework and the self-managed cell. Finally, it considers the conflict resolution of policies that is expected to occur in the Y-Comm network environment.

Chapter 4: Framework of the security management system for 4G heterogeneous networks

Chapter 4 first defines the problem that this research aims to solve, then discusses the design considerations affecting the proposed system. It next presents an overview of the framework with details of each part, including the management layer, IA, security engine, security database, and security administrator. Finally, it explains the sequence of messages that provide a sequence diagram for each case in the system.

Chapter 5: Prototype implementation of the security management system for 4G heterogeneous networks

The fifth chapter introduces a scenario for the proposed system's interaction during a malicious event. It then describes the problems that the proposed system faces in the Y-Comm network environment. This chapter also explains the policy enforcement points in the proposed SMS4HN. These include two components of the Y-Comm network: the access router and the core-endpoint server. In addition, the chapter describes the policy feedback loop that illustrates the interaction of the SMS4HN components with the Y-Comm network components. It presents the SMS4HN specifications with details of the

obligation and authorisation policies, and shows how the SMS4HN deals with the authorisation policy conflicts that may occur in the Y-Comm network environment. The chapter also explains how the managed objects communicate, and how to deal with exceptions and errors that may occur in the SMS4HN. Next, it describes a way to send messages to the managed objects, so that they can be controlled. Finally, it shows snapshots of the SMS4HN's responses to events.

Chapter 6: Case Study

Chapter six reports a case study of the practical application of the proposed SMS4HN. It first describes the scenario, then explains the vertical handover of an EUD in the Y-Comm network. Its explanation of how the proposed SMS4HN works in the case study includes details of the policy authorisation framework, policy enforcement points, and the self-managed cell. It next describes the IA's mechanism for detection of malicious events, using an anomaly-based detection system. Finally, this chapter explains how to create a normal model of the application's behaviour and how it monitors system calls.

Chapter 7: Evaluation of the security management system for 4G heterogeneous networks

Chapter seven describes the evaluation of the SMS4HN based on five criteria chosen to measure its ability to work in the heterogeneous environment of a Y-Comm network. For each criterion, the SMS4HN is compared with other security management systems. The first criterion is the system's extensibility, which is measured in three ways: by its ability to respond to additional malicious events, its ability to enforce policies without additional hardware, and its use of additional communication protocols. The next three

criteria are the modifiability of the SMS4HN, its self-management, and its interoperability, in other words its ability to work with the Y-Comm network's heterogeneous components. The chapter then evaluates the scalability of the proposed SMS4HN and ends by considering some limitations of the system.

Chapter 8: Conclusion and future work

The final chapter summarises the research and makes suggestions for future work.

1.10 Design considerations

This final section of the introductory chapter discusses the design considerations arising from the need for this research project to address the particular nature of the Y-Comm network architecture. Chapter 2 explains that a Y-Comm network is a convergence of both wired and wireless networks. This convergence increases the challenges in creating a system for this environment.

The design of the proposed security management system involves communication and interoperability with components in the Y-Comm network architecture and EUDs. Components such as access routers and core-end point servers are diverse and may contain different operating systems, but Ponder2, which is the tool used in the proposed system, has been used in previous heterogeneous distributed systems and has worked in such environments [10, 11].

The following is a list of the five major design considerations, corresponding to the evaluation criteria mentioned in the above summary of Chapter 7.

- Extensibility:

  An important characteristic of the proposed system is extensibility. Because the Y-Comm network contains a large number of components, the policy-management system should be extendable to include new functionalities. The proposed system uses a self-managed cell (SMC) network design to meet the extensibility requirement. An SMC, as explained in Chapter 3, increases the system's ability to interact with new resources in the heterogeneous environment.

- Modifiability:

  A second important consideration is the ability to modify the behaviour of the policy-based system while it is running, without needing to stop the system for recompile or restart, because the Y-Comm is expected to have connection services that should be available all the time.

- The self-management of the policy-based system:

  Self-management is also important for the SMS4HN, to reduce the need for human input, which would be too slow to match the rapid changes in the Y-Comm network. The self-management of policy-based system means responding to malicious events instantly without the need for human involvement. This leads to more efficiency and better security for the proposed SMS4HN.

- Interoperability:

  Interoperability with the heterogeneous components of the Y-Comm network is another important design consideration, because Y-Comm has many different

components and technologies with all of which the SMS4HN needs to be able to interact.

- Scalability:

  The scalability of the SMS4HN is an important consideration, because the Y-Comm is a large network and is expected to increase in size.

Chapter 2: Background and Literature Review

Objectives:

- Provide an overview of mobile computing and 4G;
- Provide an overview of Y-Comm;
- Provide an overview of policy;
- Discuss policy specification languages;
- Discuss similar policy-based security management systems.

2.1 Overview of mobile computing

2.1.1 Evolution of mobile network generations

Mobile networks have developed rapidly. There is always further demand for other uses of mobile devices. These demands encourage innovations and research to provide further designs for mobile networks. Since the 1980s, when its development began with the 0G precellular mobile technology, wireless technology has gone many through significant changes. Then, in the early 1990s, 1G technology was released and mobile devices became known as cell phones [12].

At this time, the first generation of mobile communication networks used analogue signalling. Analogue systems were implemented in North America and were called analogue mobile phone systems (AMPS). Total access communication systems (TACS), another 1G technology, were implemented in the rest of the world. The analogue system is a circuit-switch technology which was used only for voice calls [13].

After 1G, the second generation of mobile networks emerged and underwent very strong growth in the number of end users, due to the better service provided. These 2G networks enjoyed the advantages of other technologies, such as compression and coding techniques, combined with digital technology. At one time, there were three main mobile systems: time division multiple access (TDMA), code division multiple access (CDMA) and global systems for mobile communications (GSM) [14].

TDMA was implemented in North America in 1993. It operated in the AMPS frequency of 824 MHz. There were also the 894 MHz CDMA systems, which used the direct

sequence spread spectrum (DSSS) and worked in the 1850–1990 MHz frequency range to support CDMA users. GSM was designed for voice services and for limited data capabilities, such as the short message service [12]. By the late 1990s, 3G cellular systems had been released to satisfy the need for a universal technology and advanced performance. In addition, 3G communication networks provided higher speeds for information transfer of at least 200 Kbit/s. These standards met those of the International Mobile Telecommunications-2000 [15]. Recent 3G versions, often called 3.5G and 3.75G, also support higher speeds up to several Mbit/s for smartphones. Subsequent versions of 3G provided further applications in wireless video calls and mobile TV. Different standards of 3G have been applied in different regions of the world. The Universal Mobile Telecommunications System (UMTS) system was introduced in 2001 and primarily implemented in Europe, Japan and China. The cell phones were hybrid and were able to work on both GSM and UMTS systems. An advanced release of UMTS, HSPA+, supported speeds up to 56 Mbit/s. Another standard of 3G, called CDMA2000, was implemented in North America and South Korea [13]. Smartphone applications increased the demand for further developments in high-speed connections and more reliable service, which were then supported by the design of fourth generation (4G) technology for mobile networks. In section 2 of this chapter, 4G will be explained in detail. Figure 2.1 summarises the evolution of mobile generation networks.

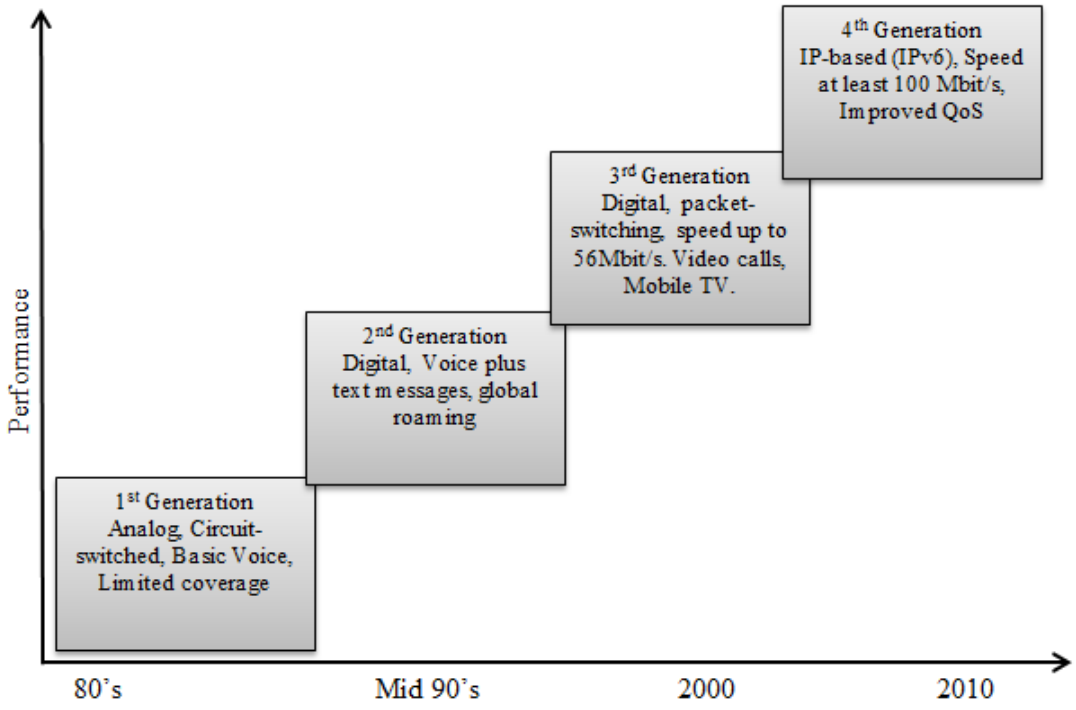

Figure 2-1. Evolution of mobile network generations

2.1.2 The challenges of mobile computing

There are challenges in designing software systems for mobile computing which are quite different from those faced in designing for stationary networked devices such as PCs. Wired communications face fewer challenges than wireless communications do. The particular challenges facing mobile computing are as follows:

- Wireless communications are affected by the environment, which interacts with signals. The surrounding environment can block the signal, make noise and

echoes, etc. Hence, wireless communication involves lower bandwidths, higher error rates and the possibility of disconnection. These factors cause communication latency as a result of retransmission, error-control protocol processing, etc. [16]. Therefore, network failure is often a greater issue in mobile computing than it is in stationary networked devices.

- Loss of some information happens in wireless networks such as 3G mobile networks more frequently than it does over wired networks, due to the ability of users to move physically to different places while connected to the network. Thus, there is need for improved technology to provide better service to end-user devices.
- Mobile devices are resource-poor compared to static computing devices. Considerations specific to mobile computing, such as weight, power and size, affect computational resources such as processor speed, battery, memory size and disk capacity. Even if the capabilities of mobile devices continue to improve as they have done, they will always be resource-poor compared to static computing devices. Moreover, concern about power consumption needs to be taken into consideration at the levels of hardware and software design [17].
- Mobility is susceptible to risks. There is the possibility of devices being lost, stolen and damaged as a result of weather problems, unlike stationary devices, which are typically kept in safe places.

These challenges and constraints must be taken into consideration while designing hardware or software for mobile computing. Some of them have been considered during

the design of the proposed approach to enforcing policies and collecting data through intelligent agents on mobile devices.

2.1.3 Security issues in mobile computing

In recent years, there have been major developments in mobile technology. One of the greatest concerns in such developments is security. The absence of wires provides mobility, while network access allows the use of communication-based applications. This combination of mobility and networking presents new and bigger challenges. One of the more challenging problems is security [1, 2]. The security solutions for fixed-computing devices are simple and straightforward compared with those for mobile computing. One such solution is the physical isolation of each fixed computer and database from other environments. By contrast, it is difficult to isolate mobile computing devices because of limited resources and the necessity of communication with mobile support stations [18, 19]. Hence, mobile computing is affected by two particular security issues, concerning location information and denial of service (DoS).

- Location information issues

The location of a mobile device will often constitute important data, whereas most users want their location to be kept confidential and want to remain anonymous from other networks and users [20]. There are many problems related to anonymity, including the trust afforded by the mobile support stations and the problem of sharing and transferring location data between network nodes. This requirement is more important when the user moves between different nodes (different zones) which have different levels of trust [18, 20].

- Denial of service

A DoS attack can occur in mobile networks if an overwhelming number of registration requests are sent to the visiting network. The receiver cannot detect that the requests are fake until it attempts to authenticate each request. To authenticate each request keeps the network busy and the fake requests cause the real registration request to become lost [1].

2.2 4G mobile networks

2.2.1 Overview of 4G mobile networks

The International Telecommunications Union (ITU) defines the International Mobile Telecommunications-Advanced (IMT-Advanced) standard as the global standard for 4G wireless communications. As specified by the ITU's ITU-R Recommendation, 4G provides very high speed connections, such as 100 Mbit/s for outdoor environments and 1 Gbit/s for indoor environments. In addition, it is recommended that a 4G network should have high capacity, low cost, low latency, good quality of service and good coverage [15]. 4G networks have been developed in response to the increasing need for ubiquitous connectivity and service provision [4, 21]. There are many candidates, such as Long Term Evolution (LTE) Advanced and WirelessMAN-Advanced, which seek to meet these requirements [22], especially the need for high speed, while other candidates are trying to build a 4G heterogeneous network as a convergence of wired and wireless networks. Among the new architectures designed for this heterogeneous network, Y-Comm comprises a faster core network and a slower peripheral network.

There are many differences between 4G and previous versions of mobile communication systems. One important difference is that 4G mobile networks will operate completely on the TCP/IP architecture. This design decision was made to reduce cost and to break what has been known as the closed cellular market, supported by limited service providers [6]. This study proposes a system based on Y-Comm, which will be explained in the following sections.

2.2.2 Challenges to 4G

An important goal of 4G mobile networks is to provide reliable and seamless service everywhere at all times. However, due to the high requirements of such networks, there are many challenges facing those seeking to meet these technical requirements, including cost, quality of service, billing and security.

- Cost of 4G technology

Although 4G mobile networks use a higher frequency band to achieve higher speed data transmission, reducing the costs of using 4G is an important demand which has increased the challenges to 4G [23].

The problem is that using higher frequencies reduces the radius of coverage of each base station [5]. To overcome this, providers need to increase coverage while providing higher frequencies. Among the technologies used to address this are improving the modulation and demodulation techniques, and using adaptive array antennae and low-noise receivers [5, 23, 24]. However, these attempts to overcome this challenge have not fully satisfied the requirements of IMT-Advanced in 4G networks, as explained above in section 2.1.

- Quality of service

Another challenge to 4G mobile networks is quality of service (QoS), which raises many concerns. First, many different services generate traffic, which leads to heavier traffic loads on the network. Moreover, transmitting the various streams associated with these diverse services, such as video, voice and data, imposes a range of different QoS requirements [12]. A second source of QoS issues occurs during handover, due to the converging of non-IP-based and IP-based networks and devices in 4G mobile networks [25]. Numerous studies, including [26, 27], have attempted to find solutions to this challenge; however, their proposed architecture needs more experimental and analytical research to ensure that the performance satisfies the 4G requirements. Moreover, end-user devices have to process signals received from various systems, discover available services and connect to the appropriate service providers. QoS is affected, because different service providers have their own various protocols, which may be incompatible with each other and with the diverse EUDs, making it difficult to provide satisfactory service to the end user [12]; however, this challenge will not be addressed in the present study, because many other research studies have investigated it.

- Servicing and billing

Another challenge to 4G is servicing and billing, which are becoming increasingly complex, particularly due to the continual interaction of service providers in allowing EUDs to access their networks. Several studies have investigated this challenge and proposed frameworks to deal with billing and end-user account information [28-30].

- Security and privacy challenges

Park and Park [3] showed that 4G networks suffer from many security threats, leading to service interruptions and disclosure of information, because 4G is IP-based and the networks are open and heterogeneous. They used X.805 to analyse the security of heterogeneous networks. The X.805 standard was developed by the ITU as a systematic tool on the Bell Labs Security Model [31]. Such high numbers of security threats are not seen in closed environments (e.g. PSTN), which are better protected against security threats. Park and Park obtained their findings by analysing the security threats to international communication networks such as IEEE, WiMax, 3GPP and ITU, which are integrated and converged to compose the 4G heterogeneous architecture. Mahdi et al. [4] also report that 4G heterogeneous networks suffer from more security threats than do closed networks. They examine the possibility of using 3G security techniques such as AKA for 4G networks. These threats are not seen in 3G networks, because the whole network is owned by a single operator. Consequently, there is a need for more research into the security of 4G networks. In response, the present study proposes a solution to such security concerns in a potential architecture for a 4G heterogeneous network (Y-Comm).

2.2.3 4G heterogeneous networks

End users demand high-speed network connectivity that supports multimedia services, such as video calling and voice over IP (VoIP). This demand is more important when dealing with wireless networks than wired networks, because users require wireless connection services everywhere. The bandwidth in wireless networks remains limited, creating a need for the development of efficient ways to satisfy these demands among rapidly increasing numbers of users. One proposed solution is the 4G heterogeneous

network. Heterogeneous networks are a convergence of all different and wireless networks in a common core IP-backbone which supports the interoperability of components of the network to provide a ubiquitous networking environment [27, 32]; this requires the availability of the service to be increased to meet the 4G requirement of high-speed connection anywhere at any time.

This environment will enable mobile devices to move through networks with fewer disruptions such as lost or slow connections. It provides the best available network based on the location of the mobile device. Each network in the heterogeneous network is deployed by a different service provider [33]. This study investigates an improved Y-Comm architecture for 4G heterogeneous networks; the next section offers an overview of Y-Comm.

2.3 The Y-Comm framework

Y-Comm framework was introduced by a group of researchers from the Networking Research Group at Middlesex University, the Computer Laboratory at Cambridge University, Samsung Research and Deutsche Telekom. The objective of this framework is to address new challenges in heterogeneous networks. These challenges are found at all levels: network, device and application. The framework maintains a layered approach and performs as a reference model, like the Open Systems Interconnection (OSI) reference model [34]. This research proposes a security management system for the Y-Comm framework; therefore, it is important to explain the framework thoroughly.

2.3.1 An overview of Y-Comm

Y-Comm is an architecture for mobile heterogeneous networks composed of two frameworks: the peripheral and core frameworks. The core network contains wired technologies, such as optical networks, while peripheral networks consist of wireless technologies such as 3G [35, 36]. This convergence between wired and wireless networks represents a forthcoming telecommunication environment aiming to support heterogeneous devices, different network technologies and service providers.

The Y-Comm framework is illustrated in Figure 2.2, which shows shared base systems (in blue) containing the network abstraction layer and the hardware platforms. The function of both core and peripheral frameworks is to provide services to the heterogeneous environment.

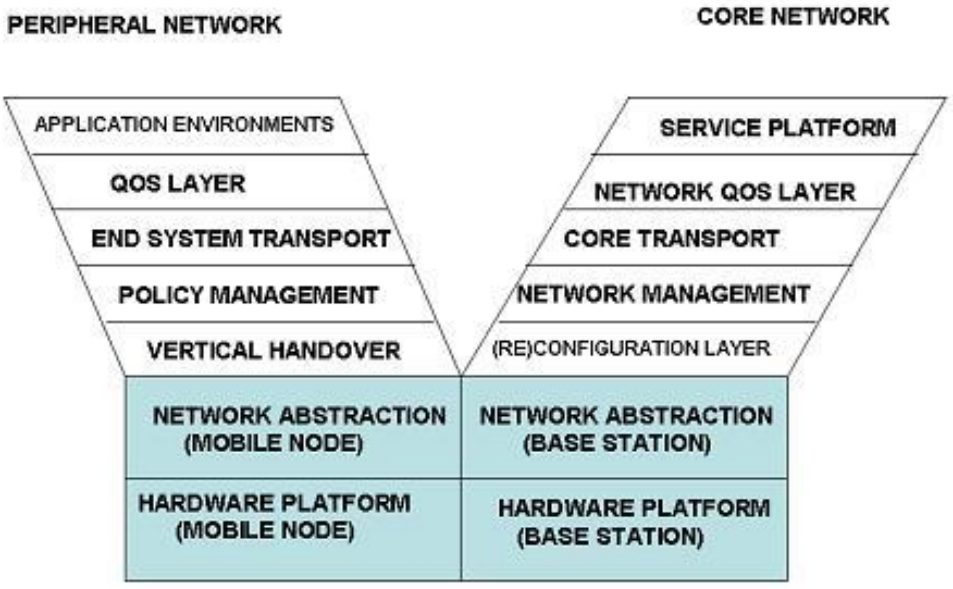

Figure 2-2. The Y-Comm framework [37]

2.3.1.1 The peripheral framework

The peripheral framework deals with mobile devices and other devices connected to wireless networks. It is composed of seven layers [37]:

- *The Hardware Platform Layer* (HPL) is responsible for the classification of all of the various wireless technologies required for the network, including the electromagnetic spectrum, MAC and modulation techniques, which compose the HPL.
- *The Network Abstraction Layer* (NAL) has an interface able to maintain and control the various wireless technologies. HPL and NAL have similar functionalities for the core and peripheral frameworks. In the core framework, they are used to regulate the functions of different wired networks, whereas in the peripheral framework, these two layers work on a mobile terminal to support the diverse wireless network technologies.
- *The Vertical Handover Layer* (VHL) performs vertical handover, which occurs when an EUD switches its connection to a different network technology or a different level of the current network's hierarchy [38]. Vertical handover is achieved by a client-based handover approach. Therefore, it is controlled by the mobile device. After handover, VHL executes the initial handover signalling, context transfer and packet reception.
- *The Policy Management Layer* (PML) determines whether, when and why handover should happen, taking account of many factors such as signal strength. It follows policy rules to determine the appropriate time and area for executing the handover.

- *The End Transport Layer* (ETL) allows the mobile device to make end-to-end connections through the network. It supports the functionalities of the TCP/IP module for both its network and transport layers.
- *The Quality of Service Layer* (QL) provides two techniques for handling QoS: downward and upward QoS. Downward QoS is used when an application defines the QoS that it needs and the system responds by managing this QoS over different network channels, whereas in upward QoS, the application is responsible for adapting to changes in QoS. Moreover, QL monitors the entire wireless network, to guarantee reliable connection.
- *The Applications Environments Layer* (AEL) defines a group of objects, functions and procedures to develop applications that can utilise the framework [39].

2.3.1.2 The core framework

This subsection is concerned with the functions performed by the core framework, which as noted above shares its first two layers with the peripheral framework. The remaining layers are the following:

- *The reconfiguration layer* (REL), which maintains the main infrastructure parts, such as routers, switches and access points, using network management applications.
- *The network management layer* (NML) is responsible for managing and controlling network operations in the core framework. It also collects information about locations of wireless networks and sends this to the PML.

- *The core transport system layer* (CTS) moves data across the core network.
- The network QoS layer (NQL) deals with QoS issues in the core network and QoS issues that occur between the peripheral and core networks.
- The service platform layer (SPL) permits the services to be set up on different networks at the same time [39].

2.3.2 Y-Comm security framework

The Y-Comm research group deploys a multilayer security model which works on both the core and peripheral frameworks. Figure 2.3 shows the complete Y-Comm framework with its security model, which is composed of layers that work simultaneously to ensure complete integration with the core and peripheral frameworks.

Because of the open architecture of Y-Comm networks, the security model is responsible for protecting the network entities and not just the data. The security model monitors and maintains these entities to ensure they are not harmed or used by malicious users.

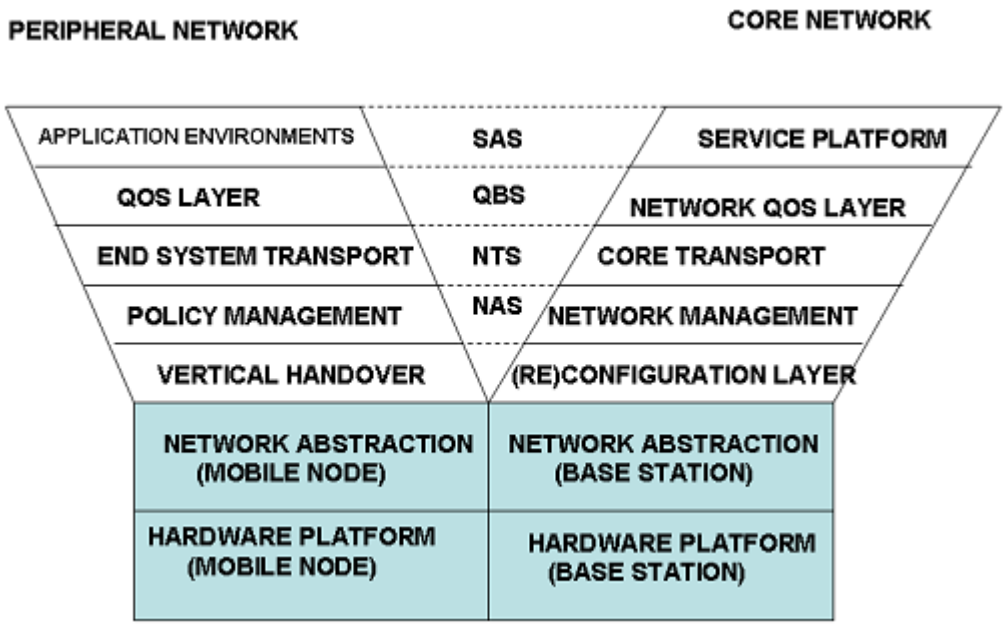

Figure 2-3. The full Y-Comm architecture [26]

At the top is the service and application security (SAS) layer, which is concerned with the authentication, authorisation, accounting and cost (AAAC) functions. It is used in the peripheral framework to authenticate users and applications, while in the core framework, SAS is responsible for the AAAC functions of the service platform.

Underneath the SAS is the QoS-based security (QBS) layer, which deals with QoS issues and the different QoS needs of the mobile environment when users move across the network. QBS also meets the service-level agreement by ensuring that peripheral networks are not overloaded due to replications of services from servers. Additionally, the QBS layer protects the network from QoS-related attacks such as denial-of-service attacks. The next security layer is network transport security (NTS). In the peripheral framework, NTS maintains access to and from mobile devices and the availability of these devices and services on the Internet, whereas in the core framework it is

responsible for maintaining secure connections throughout the network. The final layer is network architecture security (NAS), whose functions differ in the two frameworks. In the peripheral framework, NAS addresses the security threats arising from using a new network technology and ensures that each mobile device is authorised to use a network in the framework. In the core framework, NAS works to provide secure access to the infrastructure and helps the network management system to decide which switches and routers can be used [26, 39].

2.3.3 Analysis

The open nature of 4G means that the infrastructure is accessed from many external connection points through peer operators, through the Internet, and via third-party technologies. All of these elements are at risk of providing holes in security and vulnerabilities. In addition, multiple service providers share the core network infrastructure, which means that one single provider being compromised affects the whole network infrastructure [3]. The Y-Comm research group provides a security model, as explained in the previous section. However, when Aiash et al. addressed the security challenges to 4G systems by looking at the possibility of applying current security techniques to 4G networks, their results indicated that both current and new security threats were inherent to 4G technology [4]. Their study used standard X.805 to investigate the possibility of applying the 3G Authentication and Key Agreement (AKA) to a 4G communication framework. By applying X.805, they analysed the AKA protocol in 4G networks and identified many threats to the security of 4G networks [4].

An earlier study by Park and Park [3] showed that because 4G is an IP-based heterogeneous network, there were a number of security threats which could cause service interruption and permit data to be hijacked. They also addressed several outstanding open issues which required solutions. In a traditional network security procedure, the network is secured by preventing threats from accessing network entities. However, this is inefficient with an open architecture network such as 4G, because the attackers will try to find security vulnerabilities in the operating system and in the network protocols or applications, allowing them to create malware which abuses the network. According to the new architecture, possible threats within a 4G network system are: IP address spoofing, user ID theft, theft of service (ToS), denial of service and intrusion attacks [3]. Because of the open architecture and IP-based environment, 4G heterogeneous networks are vulnerable to new security threats and inherit threats from the Internet which were unseen in 3G because the network infrastructure was owned by the service providers and access was denied to other network equipment. Moreover, the diversity in EUDs and security levels leads to greater security threats [6]. The experience of Internet protection suggests that it should involve not only data but also entities, which leads to the belief that in 4G, both the entities and infrastructure should be protected [3]. Another security problem occurs in mobile communications when the EUD is disconnected from the network for reasons such as battery exhaustion. The transition from the level of disconnection to connection presents an opportunity for the attacker to show himself as a mobile device or a mobile support station [18]. There is an increasing need to protect EUDs, due to the increasing danger of rootkits. Hoglund and Butler [40] define a rootkit as “a kit consisting of small and useful programs that allow an attacker to maintain access to ‘root,’ the most

powerful user on a computer. In other words, a rootkit is a set of programs and code that allows a permanent or consistent, undetectable presence on a computer". This type of malware can modify operating system code and data for malicious reasons. McAfee stated in 2010 that rootkits had increased by 600% in the last few years [41] and that most malware targeted Android operating systems [42]. In addition, new end-user devices are sources of DoS attacks, viruses, worms and so on. Smartphones have become attractive targets for attackers and this makes the social implications of the attacks more harmful. Some security solutions, such as Y-Comm and Hockey, have been proposed for 4G heterogeneous mobile networks. However, they do not take into account the security of EUDs, which cause many security vulnerabilities, and they do not achieve the security requirements of 4G systems. In response to these shortcomings, this research study proposes a security management system for the Y-Comm network. The following chapters discuss the proposed system in detail.

2.4 Overview of policy

This research provides a policy-based system to cover gaps in security models for heterogeneous networks. Therefore, it is important to explain the concept of policy and how it can be beneficial for protecting heterogeneous networks. This section provides a definition and categorisation of policies, then attempts to explain how a policy-based system could be the best solution as a security management system.

2.4.1 Policy definition

There are many definitions of *policy,* due to the increase in complexity of management, and this can be seen especially in network management, which is becoming more

complex due to the various heterogeneous systems and technologies [43]. The definition of policy given in [44] is as "rules to control the behaviour of the system", and this definition can be applied to many systems, especially a heterogeneous network, which needs rules to govern the behaviour of its network entities and interactions with EUDs. However, [45] defines policy as a set of rules, and finds that each policy contains a condition statement and an action statement. This definition relies too precisely on one type of policy: event-condition-action (ECA). Although Strassner's definition suits some requirement of policy-based security management systems, such as business policy systems, it does not include other security policies, such as authorisation policies. There is an important distinction between authorisation policies and obligation policies. Obligation policies have two main parts: condition and action. The condition is triggered by an event, and when the condition applies, the action should be performed. Obligation policies are used to adapt the behaviour of the system, while authorisation policies are used to specify which resources or services in the network the user can access. Thus, security management policies are required to detail what actions should be performed when security violations happen [44].

2.4.2 Motivation of policy-based systems

This section justifies the use of a policy-based system. The convergence between wired and wireless technologies in 4G heterogeneous networks and the diversity of network technologies make network management more complex. It is this increased complexity which prompts this research to find an appropriate and adequate solution.

Policy-based management systems have become promising for controlling heterogeneous networks [46]. There are many reasons for the recent interest in policy-based management, including:

- It supports dynamic changes of system behaviour without the need to stop the system [46]. This feature suits heterogeneous network services, which should be available at all times without stopping or reconfiguration.
- It requires less human effort to administer the network. This is an attractive characteristic in managing large-scale networks that contain diverse network technologies, such as 4G. Therefore, it is important to produce a policy-based system that can be cost effective for the end user as well [45].
- It defines the behaviour of large-scale networks or distributed systems. With the rapid increase in the number of network users, a growing number of applications and services required by end users have been deployed, creating a need to define the rules for using such services and to control the relationships among diverse network entities. Hence, it is difficult to build a management system. Policy-based systems can help to define policy rules and to enforce them en masse [45].
- It provides better security. As very many network resources are joined in the core network of a heterogeneous network, it is essential to protect them from abuse. These resources can be abused by authorised users who misuse their network privileges. The worst case is when malicious users try to attack the network resources [45].

2.5 Security policy overview

Computer security, in general, is defined to include three main objectives of security: (1) confidentiality, which is concerned with the disclosure of information; (2) integrity, which is concerned with changes of information; and (3) availability, which is concerned with denial of access to information [47]. These main objectives are achieved through three mechanisms: authentication, access control and audit. Authentication is concerned with giving legal users access to resources, whereas access control limits what legal users of the system can use in each part of it, to ensure that resources are accessed only by authorised users. The access systems are composed of three main components: subjects, targets and rules that define which subjects are allowed to access the target in specified ways [48, 49]. Figure 2-4 shows the security objectives, mechanisms and policy types.

2.5.1 Security policy types

This research recognises that the following policies are used as the standard; therefore, they will be used as the basis for this study.

- **Authorisation policies**

An authorisation policy specifies what activities a user can or cannot do in the system [50]. Authorisation policies which allow a user access are called positive authorisation, whereas negative authorisation prevents a user from doing an action to objects of the system [45]. Note that the use of positive and negative authorisation policies may cause conflicts. However, the policy specification language which is used in this study

provides support to solve such conflicts. We explain this kind of conflict and how to deal with it in the following chapters.

- **Obligation policies**

An obligation policy specifies what the subject in the system must do if an event happens. Thus, predefined events trigger such a security policy to execute actions. This is the basis of ECA. The event can be detected with the support of external entities, such system agents in the target [45, 51]. Obligation policies have many applications, especially for dealing with security violations. When a security violation happens, a set of actions are taken to protect the network. This study uses such policies to deal with a predefined security violation. The set of actions are based on ITU-T recommendations, which will be explained in section 2.6. The policy specification language used in this study, Ponder2, supports the application of obligation policies to heterogeneous networks.

- **Refrain policies**

Refrain policies are used to prevent subjects from doing a set of activities in the system, regardless of whether they have been authorised to do so. They differ from negative authorisation policies in one way: they are subject-based, whereas negative authorisation policies are target-based [45]. Refrain policies can be used to protect a user who has suffered a dangerous attack. This study does not consider refrain policies, because dealing with security violations that may affect network objects is beyond its scope.

- **Delegation policies**

Delegation policies define the relations among the subjects of the system, whereby one subject grants a set of privileges to another. Delegation policies are associated with authorisation policies but are subject-to-subject policies [52]. This research study does not consider them.

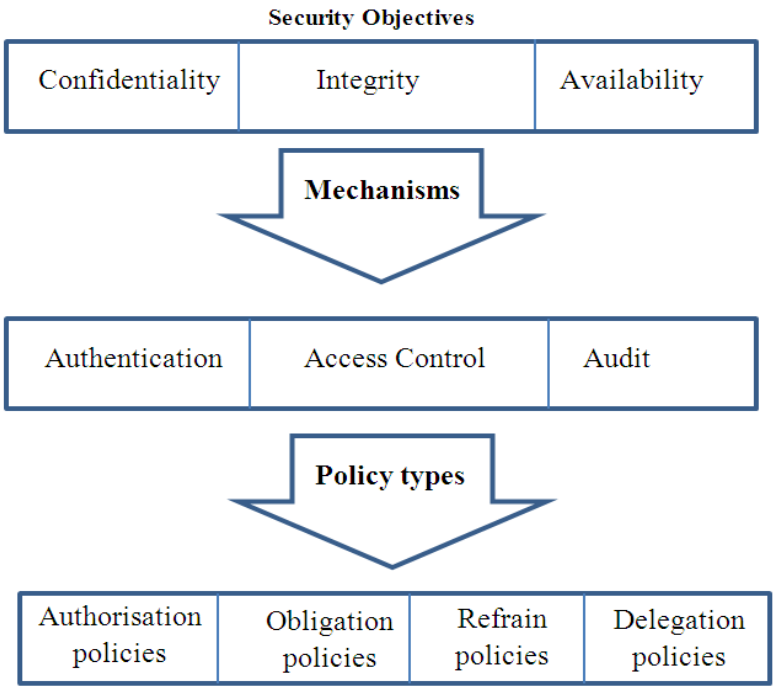

Figure 2-4. Security objectives, mechanisms and policy types

2.5.2 Access control models

Many research studies divide access control policies into discretionary and mandatory ones. Mandatory policies are responsible for controlling information flow between the objects of the system, while discretionary policies specify control of users' access to

objects. Recently, many studies have considered role-based access control policies as an alternative to discretionary and mandatory policies [47, 53].

2.5.2.1 Discretionary policies

Discretionary access control policies are responsible for controlling the access of subjects to objects of the system, based on their identity or groups. A basic mechanism of discretionary access control policy uses an access matrix model. However, a recent extension of the access matrix model has been to use positive and negative authorisation with it [48]. Negative authorisation defines when access should be denied, and it can be used to remove access rights temporarily when a positive authorisation is in place. This research uses discretionary negative authorisation to remove the access rights of a user whose privileges have been stolen with the purpose of using them to attack the network. This will be explained in more detail in the following chapters.

2.5.2.2 Mandatory policies

In a mandatory model, access control policies are based on fixed rules mandated by a central authority. These access control policies are performed using the Bell-LaPadula lattice-based model [47], which deals with issues of data confidentiality; however, such policies are not considered in this research.

2.5.2.3 Role-based access policies

Role-based access policies (RBAC) follow different ideas from discretionary and mandatory policies. In RBAC, permission to access objects of the system is specified in

rules which system users are given, so that they can access these object in terms of the said rules. RBAC is treated as a separate model in traditional studies such as [54, 55].

2.5.2.4 Other models

Additional models have been proposed to be applied in different environments, but this research does not consider them in designing and implementing the proposed security management system. Examples of other models are non-discretionary policies and the Clark-Wilson model.

2.6 Security management

A considerable amount of literature has been published on access control and security models, but few studies have investigated security management systems, especially those used in environments such as heterogeneous networks, where the different security levels make designing the system very difficult. The security measures used for wired networks might not suit wireless networks. In general, security management is considered part of network management and one of the functional areas of the OSI management framework; Langsford defines security management as being "not just related to using encryption or authentication mechanisms but also including reports of any malicious attempts to harm the system" [56]. Hence, the security management system is responsible for important issues which detect and deal with security violations. Moreover, the security management system is expected to provide a secure way to manage the network [57]. Security management is alternatively defined as "the translation of the authorisation policies to useful information that can be used as security techniques to control access or monitoring of the security activities" [58].

Another important definition, adopted in this study, is that of the Internet Engineering Task Force (IETF): security management systems deal with security policies, which are concerned with the authentication of users and granting or denying users' access to objects of the system [59]. In the proposed model, the security management system is responsible for granting or denying a user's access if a security violation happens.

Important aspects of the definitions discussed above, such as detecting a security violation and how to deal with it, lead to the conclusion that security policy is based on ECA, or obligation policies. Denying access to network resources requires authorisation policies. These two kinds of policy, authorisation and obligation, have been considered and will be explained further in the next section.

2.7 Policy specification languages

Having reviewed the concepts of policy and policy-based systems, this section now reviews some policy specification languages currently popular in various domains, such as distributed systems, network management, and ubiquitous computing. In particular, this section provides justification for using Ponder2 as a tool in this study.

2.7.1 Ponder

The Ponder policy language was proposed at Imperial College London as a result of research over 10 years. Ponder is a declarative, object-oriented programming language. It supports the specification of security policies concerning access control mechanisms, such as firewalls, operating systems, and applications [52, 60]. Ponder can be used for security management activities and it can deal with security violations. It supports four

types of policy: obligation, authorisation, refrain, and delegation. It also supports domains which are beneficial for large distributed systems, so it can group objects based on their geographical locations, types, or object authorities. Ponder is not considered in this research, because the second version, Ponder2, has improved features and provides better performance, as explained in later sections.

2.7.2 PDL

The policy description language (PDL) is a declarative, expressive language used to specify policies. PDL supports obligation policies which take the ECA format [61, 62]. The main drawback of this language is that it does not support authorisation policies, which means that it cannot be used for security management purposes, especially for the current network technologies. Therefore, this policy language will not be used in this study.

2.7.3 XACML

Extensible access control mark-up language (XACML) is a declarative policy language implemented in XML. It is mainly used for access control policies in distributed systems [63, 64]. XACML specifies the subject, object, condition, and action in XML format. It provides role-based access control, so roles and groups are specified as a set of attributes. It has two main components: policy enforcement point (PEP) and policy decision point (PDP). These components can be attached to an application or distributed systems across the network [65, 66]. Ferraiolo et al. [67] identify two drawbacks of XACML. One is that it specifies or enforces policies which are meant for isolated users; thus some of the policies required in security management cannot be enforced, such as

isolating the authenticated user when security violations happen. The other is that PDP in XACML is stateless, which further limits the specification of some policies.

2.7.4 LaSCO

The language for security constraint on objects (LaSCO) provides a graphical approach to specifying security constraints on objects. In LaSCO, a policy is represented as a graph, where objects are depicted as nodes, and events as edges [68, 69]. Although LaSCO provides an attractive graphic way to specify policies, it has a few drawbacks which make it unsuitable for security management purposes. First, it does not support obligation policies, nor can it specify policies for groups of objects. Secondly, it usually needs a textual version to specify details which cannot be explained in the graphical format. This drawback leads to difficulty in using the language in further restriction policies [44].

2.7.5 Tower

Tower is a language built to support the specification of role-based access control policies. Its basic components include the main structures of role-based access control, such as roles, users, and permissions. The language is flexible and supports approaches for tracking actions that have been executed based on past events. However, it too has a number of drawbacks; for example, it supports only a basic policy specification of security management, which makes it unsuitable for advanced and complicated systems. Moreover, it does not provide a solution for conflicts between policies, which normally happen in large-scale networks [70]. This means that Tower is not an appropriate policy

specification language for 4G heterogeneous networks, which are known for their great size and complexity.

2.7.6 Ponder2

Ponder2 is the second version of Ponder (reviewed above), developed to overcome its drawbacks and to facilitate the improvement of systems and networks. Ponder2 is a policy system which is appropriate for many environments and applications. It supports flexibility and extensibility, and provides interactivity for users to interact with the managed system. Its most important feature is that it is able to work with various software and hardware components in a wide range of environments, such as local area networks, wide area networks, and distributed systems [10, 71].

According to [72], Ponder2 is implemented as a self-managed cell, which can be any hardware or software component that is able to perform the required functions automatically. The SMC has a self-management feature and is composed of an administrative domain in the managed system. Ponder2 implements the policy-based system by treating every part of the managed system as a managed object. Managed objects can be anything, including sensors, switches, routers, or end-user devices. The concept of managed objects gives Ponder2 the seamless ability to maintain the various parts of the managed system and to utilise these components for management purposes.

Ponder2 supports two kinds of policy: obligation and authorisation. Obligation policies in Ponder2 are event triggered, i.e. when events occur they execute action. Such policies can be applied for security management purposes, so when a predefined security

violation happens, the system executes the appropriate action to handle it. As to authorisation policies, these allow or deny interactions between managed objects.

In addition, an event type in Ponder2 is treated as a managed object. An event type specifies that an event can happen at any time, and it designates what information can be held with this event in a template. An event is both an instance of an event type and a managed object. The event is created by a managed object to hold a message to another managed object within the system. The first managed object sends this message, depending on a timer or the detection of some condition [10].

Ponder2 incorporates the concept of domains, which in Ponder2 are managed objects consisting of other managed objects. The main purpose of domains is to maintain policies more easily, especially in large-scale systems. This is another advantage of Ponder2 which makes it suitable for heterogeneous networks. [71]

Ponder2 is a promising policy system and has proven to be a multiuse language. It has been used in various projects in many institutions [72-74], and has been implemented on diverse devices such as mobile phones, body sensors, and robots [10]. The various research projects that have implemented Ponder2 include e-Health systems consisting of on-body wireless sensors [72], and self-management frameworks for unmanned autonomous vehicles [75].

2.7.7 Choosing a policy system

The previous section reviewed some common policy systems: Ponder, PDL, XACML, LaSCO, Tower, and Ponder2. Their different features influenced the choice of an

appropriate system for the working environment of this research. The features and drawback of each policy have been clearly explained to check their suitability for this project as a security management system for 4G heterogeneous networks. Although, as discussed before, these policy systems support the main policy types which are needed for security management purposes and they are mainly intended to manage large distributed systems and networks, those such as Ponder and PDL are not suitable for small devices. Ponder2 is different from PDL and Ponder, in that it is more flexible and extensible, which suits environments that contain diverse network technologies and operating systems. Another difficulty arising from the diversity of technologies used in such heterogeneous networks is that it can be difficult to build a policy system for such an environment. Thus, Ponder2 provides PonderTalk, a high-level configuration language which ensures that the developer of a policy system does not need to know the low-level details of the various devices. This ability makes Ponder2 an ideal choice for environments which contain a range of small devices and different network technologies. Taking account of all the above explanations and analysis, Ponder2 has been used to build our system.

2.8 ITU-T recommendation

This literature review will now consider a different aspect of this research: ITU-T recommendation M.3400. This research recognises the ITU recommendation as a guideline on to how to deal with certain malicious events and what actions should be taken to protect the network. The ITU recommendation clearly explains that security

management should follow a set of procedures after an attack. We specify policies in this security management system based on these recommendations.

As specified by ITU-T, recommendation M.3400 belongs to the Telecommunications Management Network (TMN) recommendations, providing the security management sets of specifications of the TMN management function. It considers security management to be a part of TMN management which cannot be isolated from any telecommunication network. Security management includes four groups of function sets: prevention, detection, containment, and recovery and security administration. Figure 2-5 shows the four TMN management functions. In designing our system, we followed the specifications of security management. These contain many function sets, but as mentioned previously, we selected those which would help us to achieve our security requirements. The following are three function sets which met our requirements:

(1) The customer security alarm function set, defined as one which "supports access to a security alarm that indicates security attacks on their portion of the network". This function set supports detection of security violations in the network.

(2) The investigation of theft of service function set, which is defined as one that "supports investigation of customer and internal users whose usage patterns indicate possible fraud or theft of service". This function set helps us to recognise when mobile equipment is being attacked.

(3) The software intrusion audit function set, which is defined as "a set that supports checks for signs of software intrusion in the network". This third set also helps to detect any violation of the network or the mobile equipment.

These three are detection function sets, and hence contain and recover detected security violations. However, M.3400 recommendations also provide containment and recovery function sets; for example, the exception report action function set, which supports action to limit security breaches and provides some mechanisms, such as isolation of the device, to ensure that corruption is not increased or spread to other devices. Another function is the ToS action function set, which supports the limiting of security breaches by removing the user's access privileges [15]. We have built our policies on these function sets and have defined procedures to be followed if a security violation happens. The procedures for dealing with security violations in terms of the policy-based system are explained further in the following chapters.

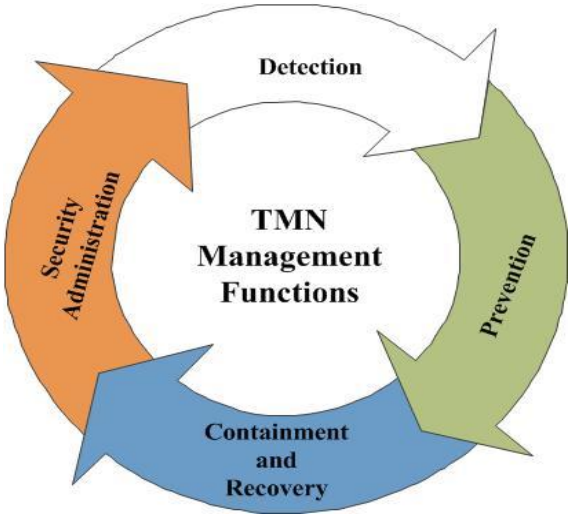

Figure 2-5. TMN Management Functions

2.9 Overview of malicious event detection mechanisms

This section presents the IA detection mechanism. As will be discussed in Chapter 4, the IA works as part of the proposed SMS4HN and is responsible for detecting an EUD's malicious behaviour that may result in compromised privileges or sensitive data theft. This research discusses the possible detection techniques, then the appropriate technique is chosen.

As a result of the openness of the Android Applications Store, an EUD may download malicious applications. An EUD may also be maliciously attacked when browsing the web. To detect malicious applications, the IA needs to understand the normal activity of an application. This can be done by creating a normal model of trusted applications. When any application in the EUD behaves outside this normal model, it is liable to harm the EUD, which may include damaging or stealing data. However, this research project focuses on stealing the user's network resource usage privileges. The user's privileges include the right of the end user to access, modify, or perform functions using network resources. These privileges are given to the end user as a part of the development of mobile networks, as explained in Chapter 3.

2.9.1 IA detection mechanism

Other research studies have proposed many mechanisms by to detect malicious activity in smartphones. One mechanism monitors the smartphone battery consumption, such as in [2], then creates a normal model of the EUD's battery consumption, against which it can detect abnormal battery usage. This mechanism is inefficient, because the end user

may infrequently use benign applications that consume more battery, such as GPS applications and games.

Another mechanism, found in [3], monitors the usage of device resources, such as RAM and the CPU. Although this monitoring distinguishes between malicious and normal applications, one major drawback of this mechanism is that it has recently experienced a high false negative rate when used for smartphones [76].

Another popular mechanism used by most antivirus software products is to build a large database of malware. This antivirus database contains signatures for each malware, obtained by analysing the behaviour of the current malwares and detecting raw patterns for each. When the antivirus software finds similar signatures, i.e. similar behaviours, in an application's operating system, it identifies this application as malware. Although this approach is common for most antiviruses, it has limitations, including the inability to detect new malicious activity in the system.

Other research studies have built anomaly-based intrusion detection systems. This approach creates a model of the application's normal activity in the operating system. It starts by monitoring the behaviour of normal applications and creating a model of normal behaviour, which allows the IA to detect strange behaviour in the operating system. Monitoring this behaviour involves monitoring the system calls requested by each application. The system calls in the Linux kernel include service application requests from the operating system's kernel.

The IA works as an intrusion detection system which is host based, meaning that the IA monitors the activities occurring in the Android OS. Therefore, the IA is anomaly

based. The IA monitors application activities in the Android OS, collects information, and identifies the normal application behaviour. Having created a normal model of the Android activities, the IA can detect abnormal activity. If it finds abnormal behaviour, it sends a report to the security engine. Section 6.5 explains how to create a normal model of application behaviours.

2.10 **Policy-based security management systems**

This section offers a review of related work and some security management techniques. The various policy-based systems use different mechanisms to manage the network and to provide a secure environment; however, this section reveals the vulnerability of some related work, showing that the Y-Comm heterogeneous environment needs an improved approach, which is what this study introduces in the following chapters.

2.10.1Automatic policy-based systems

Automatic policy-based systems have been used to manage security policies in the network automatically. Burns and colleagues designed and developed a system to reduce human involvement in network management [77]. Their system seeks to uphold security as the network changes, reconfiguring the network if necessary. They have built automatic management systems to help the systems administrators to enforce policies, because of the high number of changes in network configurations and the rapid growth of network elements, which makes managing the network difficult. The main component in their system is a policy engine which validates the policies and generates new configuration settings for network elements when policies are violated. However, there is a security challenge to this approach: how to prevent illegal users from gaining

access to the network after reconfiguration. The system proposed by Burns et al. aims to separate policy specification from the recognition of changes in the target network to add more automation to the management. An additional component is the management console, which is responsible for reporting the current state of the network and storing this information in persistent memory. However, this technique is not efficient in an environment such as Y-Comm, for many reasons. First, as explained previously, new service providers can join the core network in Y-Comm, which makes it difficult to install a management console for each network administrated. Moreover, the installing of more components will increase the cost of providing the services, which contravenes the security requirements of 4G networks, as discussed in section 2.2.

An extension of this approach is therefore proposed by Lapiotis et al., in the form of a security management system focusing on wireless network security issues [78]. They present a policy-based system architecture, which includes a central policy engine, wireless domain policy managers, and local monitors. The main motivation behind their proposal is the widespread use of wireless local area networks, which brings with it a heavy increase in security risks, in the form of malicious attacks. This research assumes that such attacks may be perpetrated not just by external attackers but from inside, by users of the network. Lapiotis et al. adopt the concept of the policy engine as the brain of the system, validating policies and generating appropriate configuration settings when access policies are breached. The distributed wireless domain policy managers are responsible for monitoring and controlling the access points. The third part of the system comprises local monitors, which contain intrusion detection modules. Policy is enforced in this system by a monitoring and instrument layer, responsible for reporting

changes in the network to the policy engine, then receiving and implementing the new configuration settings. Figure 2.6 shows the architecture of this system.

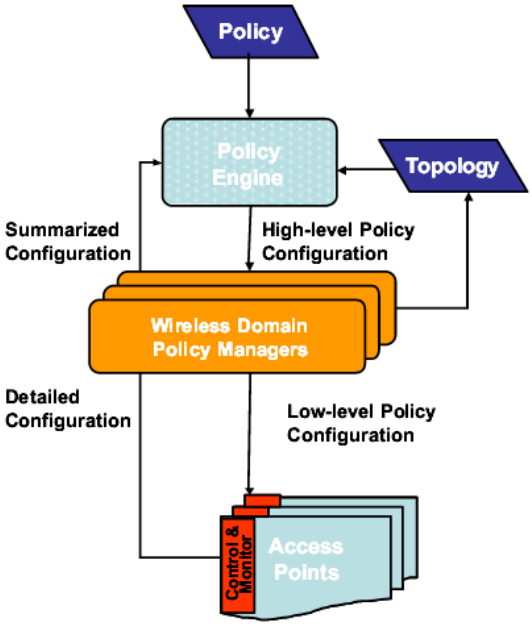

Figure 2-6. Wireless policy-based logical architecture [78]

The system proposed by Lapiotis et al. provides features such as protecting the network from new security threats without relying on the latest security mechanisms. However, one question that needs to be asked is whether detection of abnormal traffic is sufficient in considering the demand for a highly open network. Because of the high demand for highly open heterogeneous networks to provide satisfactory services to end users, such as high-speed, anywhere, anytime connection, this security management system is not suitable for heterogeneous networks. Another limitation of the proposal is that it explains neither what kind of security policies have been enforced, nor the formal validation of policies. The present study follows the same approach of a policy engine as the brain of the system, but with many modifications. Moreover, separating policy

specification from enforcement makes it more dynamic and efficient in an environment such as Y-Comm, which contains multi-layered security services.

2.10.2 Security management system based on IP address and policy zones

Other researchers have proposed a security management system based on IP addresses [79]. Their proposed system is supported by a Spatio-Temporal Role-based Access Control model. They divide the network into policy zones, so that policy enforcement can be more efficient, according to their conclusions. The main motivation behind their work is to address the many changes in dynamic, volatile wireless environments, such as increased malicious attacks and a diversity of network elements. The introduction to their model of policy zones to represent location, and the role permission given to the end user to access network resources are based on these zones. Figure 2.5 shows the conceptual framework of the security management system based on policy zones. It has six main components: home agent, foreign agent, central authentication & role server, local role servers, global policy server, and distributed wireless policy zone controllers. The local policy server is responsible for enforcing the policy in each zone. However, the need for a server in each zone will increase the cost and complexity of the management of large, diverse networks. The concept of dividing the network into policy zones is inefficient, because there will be legal network users who access the network remotely from outside the controlled policy zones. Another weakness of this approach is that it is not scalable to wide networks, or when the current network merges with other networks. They assume that the mobile IP is always specific to a host and does not change from one location to another, as mentioned in [80]. However, this is not

applicable when the network is composed of both wired and wireless technologies, as in Y-Comm.

Hence, the current research proposes a system based on the mobile ID in the service-level agreement in the core server (Section 2.3). Moreover, we are enforcing policing by using current network resources with no need for more network equipment, as explained in the next chapter.

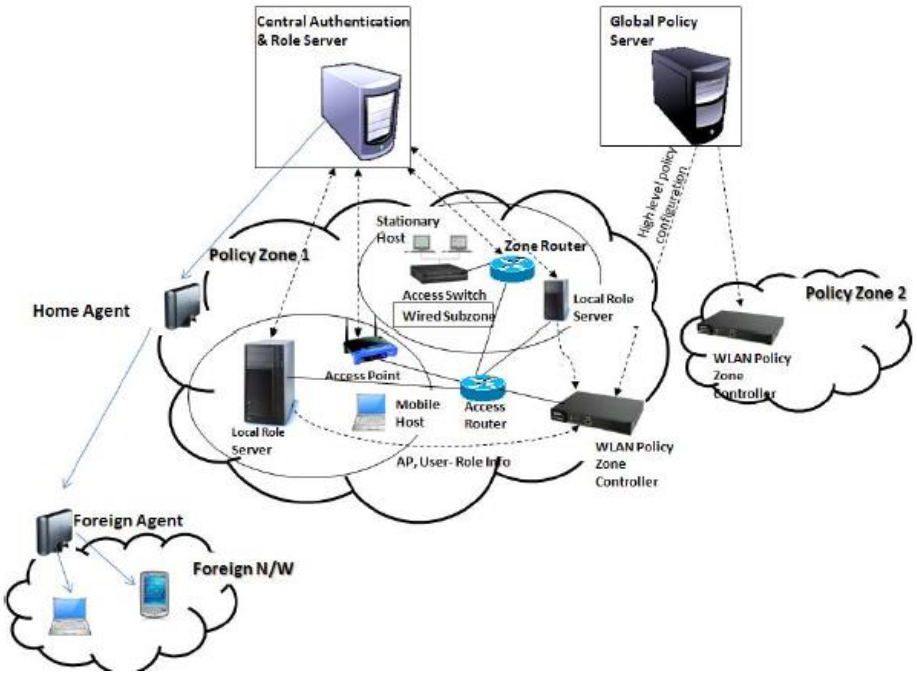

Figure 2-7. Wireless security management system [79]

2.11 **Summary**

This chapter has reviewed important topics relevant to this research, whose title indicates that it is related to networks, security, and policy-based systems. The chapter has:

- Introduced mobile networks and outlined their evolution;
- Given details of the fourth generation of mobile networks, including security and other challenges to them;
- Discussed the Y-Comm framework, with an analysis of its security models.
- Offered an overview of policies and justified the adoption of a policy-based approach to security management systems.
- Compared policy specification languages, justified the choice of policy language for this research project, and offered a critical review of related work.
- Discussed the ITU-T Recommendation, to explain why this research project follows these recommendations.

The next chapter addresses the structure of Y-Comm and related components, compared with our proposed system. Security requirements are also discussed, in line with the assumptions made by this research project. The chapter also introduces the fundamentals of Ponder2 and considers how this research project can use them.

Chapter 3: **Preliminaries**

Objectives:

- Explain the structure of future heterogeneous networks (Y-Comm);
- Present the security requirements of 4G mobile networks;
- Present the assumptions made by this research project;
- Explain the fundamentals of Ponder2.

3.1 Introduction

The primary goal of this research project is to meet the security requirements of a 4G network. The new architecture of the Y-Comm network is designed to meet the service requirements of 4G networks. However, the combination of both wired and wireless networks has raised some concerns about quality of service, handover, and security. The different security levels and heterogeneously combined networks lead to additional security vulnerabilities, motivating the proposal of a novel policy-based security management system for the Y-Comm network. There are many different definitions of a security management system, but this research chose the definition from ITU-T M.3400, as explained in Chapter 2. As the research proposes a new approach to protecting the Y-Comm network architecture, this chapter presents a more detailed explanation of this architecture. It also introduces the fundamentals of Ponder2, which is the policy specification language used in the proposed system.

This chapter is structured as follows. Section 3.2 explains the structure of the Y-Comm network. Section 3.3 presents the justification of this research project, which considers access to users' information more harmful in the Y-Comm network, leading to further attacks. Section 3.4 discusses the security requirements of 4G mobile networks. Section 3.5 presents the research assumptions. Section 3.6 explains the fundamentals of Ponder2. Section 3.7 closes the chapter by discussing policy conflict, a common problem in policy-based systems, and a resolution strategy using Ponder2.

3.2 Structure of future heterogeneous networks

Future heterogeneous networks will be owned by multiple operators, who will be able to join a core network which allows interoperability between them [81]. ITU-T recommends a central management entity that works as a regulatory authority to control the whole network [82]. This central management entity has the power to enforce policies in the network, and it contains the service-level agreements (SLAs) and network-level agreements (NLAs). The Y-Comm network adopts this concept and the proposed core-end point to work as an administrative entity that controls the peripheral networks [83]. The proposed policy-based system enforces policies in the Y-Comm network architecture using this administrative entity.

Figure 3-1 shows the structure of the Y-Comm network, with the core-end point at the top and the peripheral networks at the bottom. (There is an error in Figure 3-1, which has been acknowledged by its editors: the label ‘DCA3C’ should read ‘DA3C’.) The peripheral networks provide services to end users via an access router (AR). The middle level contains the domains, each representing a network operator. The central A3C server (CA3C) handles the central authentication, authorisation, accounting, and cost of the Y-Comm network. It contains both SLAs and NLAs. SLAs specify the clients’ terms of use of the service, while NLAs specify the clients’ terms for accessing the network [83]. The AR is the link between the network provider and the EUD. The AR is also responsible for enforcing admission control decisions. It acts as an authenticator for network users after receiving permission from the CA3C server in the core-end point. Other components deal with other issues in the network, such as QoS and handover.

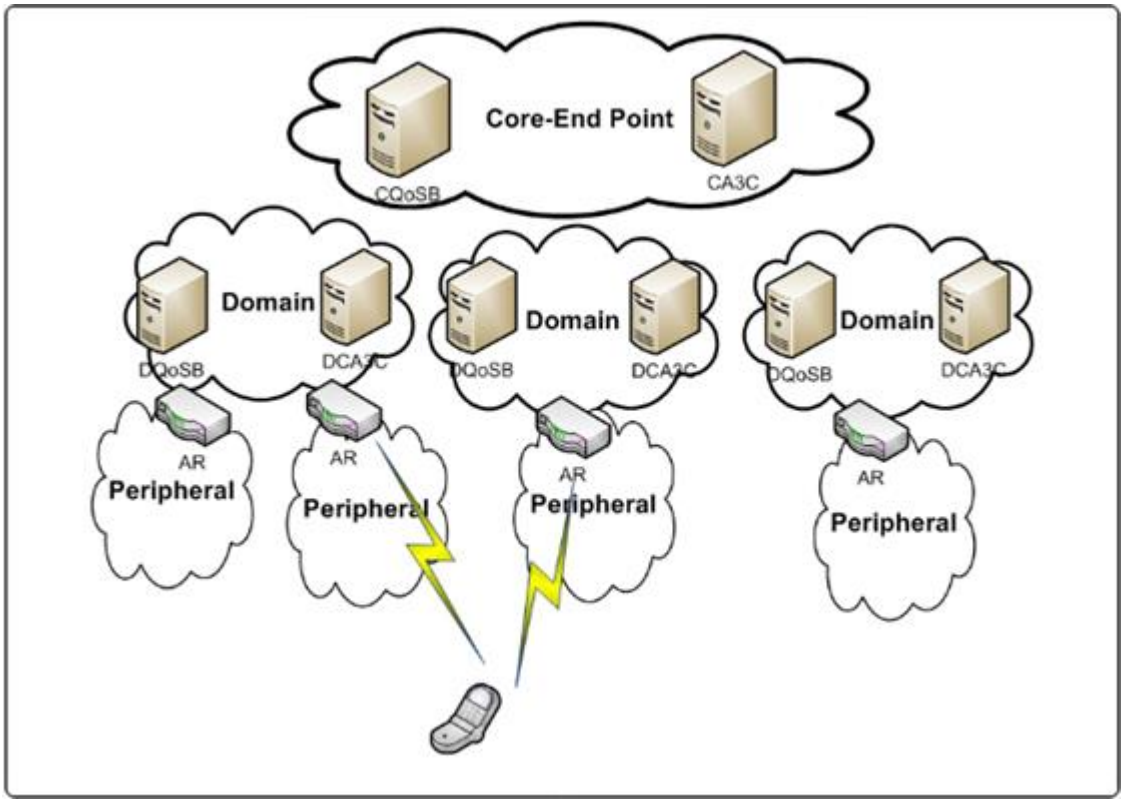

Figure 3-1. The core-end point structure with the attached networks [83]

The proposed SMS4HN interoperates with these components, as explained in later chapters.

3.3 Justification of the danger of access to a user's identity

This section addresses a question fundamental to this research: Why does this research project consider access to users' information more harmful in the Y-Comm network? User identity theft is becoming increasingly risky in mobile networks, because their users participate in ever more activities. Guo et al. [84], who studied smartphone security as part of a Microsoft project, explain that smartphone identity theft is common. The danger occurs when the attacker behaves like a normal user in the

network after stealing a user's identity. The attacker may then harm the network's resources. This danger reveals the need for a network security solution that detects malicious behaviour and takes action to protect network resources. In [85], the researchers investigate the security of smartphones and conclude that some malware harms not only the device itself, but also other elements of the network. The danger increases if the malicious attacker takes full control of the EUD. Therefore, an integrated solution with a network security model is necessary. We believe that the increase in user privileges on the network due to additional applications makes stealing users' identities more harmful to the network. The attacker who attempts to steal a user's identity may attempt other attacks using this identity [86].

3.4 Security requirements of 4G mobile networks

This section reviews the security requirements of 4G mobile networks. The present research aims to meet a security requirement that has not been clearly met by existing Y-Comm network security models. Furthermore, the proposed SMS4HN is extendable to achieve additional security goals.

Many research studies [5-7] have investigated 4G networks and concluded that any future mobile network should meet important security requirements. These may have similarities with other fields in wireless networks or distributed systems, but would improve users' involvement in network applications and resources. The Y-Comm network architecture should meet all the necessary security requirements to provide a secure environment for the user and the network. The security models for the Y-Comm network were explained in Chapter 2, where an analysis of the security model was also

provided, and where the security vulnerabilities of the Y-Comm network were discussed.

Zheng et al. [7] explain that not considering mobile device security will lead to increases in security vulnerabilities within 4G networks. Having investigated the security requirements of 4G networks, they conclude that these concern both mobile devices and network operators. Five security requirements apply to mobile devices: the integrity of the hardware and software of the mobile device should be protected, the security system should control access to the mobile device data, the integrity and confidentiality of data stored or transported to the network operator should be protected, the security system should protect the users' identities, and the security system should prevent mobile devices from being abused and used as attack tools. This last requirement is important because of the additional privileges that users have with regard to network resources. It has been investigated and met using the proposed security management system.

The security requirements for network operators are explained in detail in [7]. They have also been addressed in the Y-Comm network security model, as discussed in Chapter 2.

3.5 **Assumptions**

This section presents the assumptions made in this research project. The Y-Comm network is developing rapidly every day and has components that are not yet available to the public because of their complexity. Thus, this research makes the following assumptions:

1- That the components of the Y-Comm network are black boxes.

2- That the proposed framework will be implemented in the Y-Comm network architecture.

3- That CA3C provides the interface to interoperate the customers' terms of use for a network using an NLA.

4- That the AR for the Y-Comm network structure is integrated with a policy-based system which operates within the management system.

5- As explained in Section 3.2, that the Y-Comm network follows the ITU-T recommendation, which identifies the need for a central administrative entity responsible for enforcing policies in an environment that contains a large number of objects.

6- That the proposed system's components are trusted and able to enforce policies in such an environment.

7- That the IA is scalable and will work with different EUDs.

8- That these EUDs are allowed to contain software which works as part of the network security system.

9- That the security administrator component of the SMS4HN is automated and supports dynamic policies. This component is explained in section 4.4.2.3.

Figure 3-2 shows how these assumptions lead to the SMS4HN as presented.

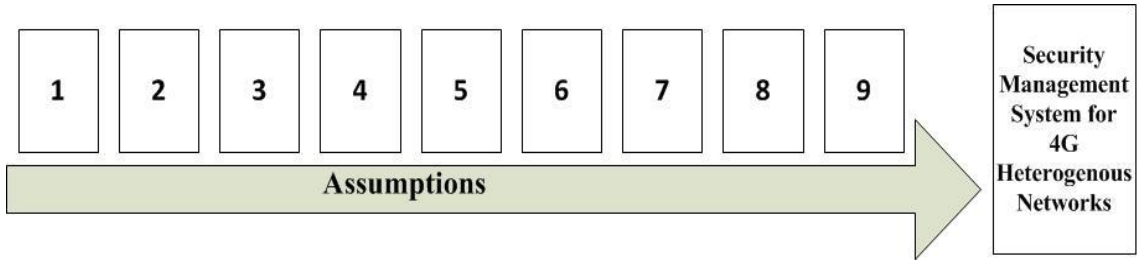

Figure 3-2. The assumptions of the SMS4HN

3.6 Problems facing SMS4HN in the Y-Comm environment

This section discusses the problems that the proposed SMS4HN faces in the Y-Comm network environment. These are overcome by applying techniques such as self-managed cells and managed objects.

Due to the increasing popularity of 4G mobile networks and the need for high speed connections, the Y-Comm network has competed strongly for 4G service requirements, including a superfast connection and the network's availability anywhere at any time.

The list of problems that SMS4HN faces in Y-Comm are as follows:

- **Convergence of wired and wireless networks**: The nature of the Y-Comm architecture, which is based on the convergence of existing wired and wireless networks, increases the difficulty of implementing a security management system to protect the Y-Comm network.

- **The security management systems** used in a wired network do not suit wireless networks, due to the hosts' dynamic topology and mobility.

- **The open nature of Y-Comm** allows new network providers to join the core network and provide connection services to end users. Thus, the proposed SMS4HN creates managed objects to deal with the Y-Comm network components, so policies can be easily enforced. These managed objects interact with the security engine (the system's 'brain'), regardless of the components' configuration details.

- **Different security levels in the Y-Comm network**: Allowing for new providers to join the network leads to different security levels in the Y-Comm network, thus increasing the complexity of the system. This research project follows ITU-T Recommendation M.3400, as explained in Chapter 2, to overcome the challenge of different security management system definitions.

This research assumes that the Y-Comm network components provide interfaces that interact with the SMS4HN components. This assumption deals with the closed sources of the Y-Comm network components.

Constructively, this research uses Ponder2 to implement the SMS4HN, as it provides a concept of managed objects, making the management of the Y-Comm network resources achievable, regardless of low-level equipment specifications. However, the difficulty for the researcher of learning Ponder2 has increased the time necessary to finish this project. This research considers the minimal support for Ponder2 and its limited resources to be drawbacks to its use.

3.7 Fundamentals of Ponder2

As Ponder2 has been chosen as the policy specification language for this research, this section introduces it and explains its fundamentals.

Ponder is a policy specification language developed at Imperial College London in 2001. It includes a number of tools and services for the specification, analysis, and enforcement of policies. A large number of subsequent developments and improvements have resulted in a new version, Ponder2. The most important improvement is the extension of the framework to work not only in general network and systems management, but also in diverse embedded devices and most complex heterogeneous systems [87, 88].

Ponder2 is implemented as a self-managed cell, i.e. a representation of a set of hardware and software components which forms an administrative domain that functions autonomously and is ready for self-management, making everything in Ponder2 a managed object. Managed objects are real-world objects, such as routers, switches, servers, or alarms that can generate events and interact with other objects for system management purposes [10]. This research treats some components of the Y-Comm network architecture, such as AR, CA3C, and NLA, as managed objects, so management policies are easily enforced. Chapter 5 explains the representation of these components and how they interact with the proposed policy-based system.

The creators of Ponder2 designed it to use managed objects, because they are allowed to interact with other software and hardware components in other environments. These components range from a single device to large networks and distributed systems [10].

The basic objects in Ponder2 are events, policies, and domains. In Ponder2, there are two basic policy types: obligation and authorisation. As explained in Chapter 2, obligation policies in Ponder2 are triggered by events, so when a managed object generates an event, another managed object evaluates this event and executes an action if applicable. Authorisation policies in Ponder2 systems allow or deny messages between managed objects [10], and when the components of the system are managed objects, authorisation policies can be enforced. The subject in this research environment is the EUD. This research treats the EUD as a managed object, so the management policy-based system is able to deny a specific EUD when necessary.

PonderTalk is a high-level configuration language that controls and configures the Ponder2 system. The managed objects in Ponder2 use PonderTalk to interact [88].

Ponder2 has four main components: the domain service, obligation policy interpreter, command interpreter, and authorisation enforcement. The domain service comprises managed objects and their hierarchical structure. The obligation policy interpreter manages ECA policies. The command interpreter is responsible for compiling the commands from PonderTalk to the managed objects. The authorisation enforcement provides a framework to enforce authorisation policies [10, 88].

The SMS4HN employs the features of Ponder2 to achieve the research goals. As previously mentioned, Ponder2 works on heterogeneous networks due to the way the managed objects are built. However, the deployment of the SMS4HN faces major challenges from the multiplicity of components and layers in the Y-Comm network. An explanation of the challenges to research project is presented in Chapter 5. In addition,

the time available to work on all the components of the Y-Comm network required boundaries to be created for this project. However, these boundaries can be overcome due to the simplicity and extensibility of the SMS4HN. These design considerations are explained in the next chapter.

3.7.1 Authorisation policies in Ponder2

This subsection explains authorisation policies in Ponder2. This research evaluates the authorisation policy framework proposed by the Ponder2 creators to achieve the goals of the proposed security management system. The Ponder2 authorisation framework (PAF) provides a way to enforce authorisation policies that can protect both the subject and the target [88].

Figure 3.3 shows that PAF supports four policy enforcement points (PEPs). PEP1 and PEP4 are for the subject, while PEP2 and PEP3 are for the target.

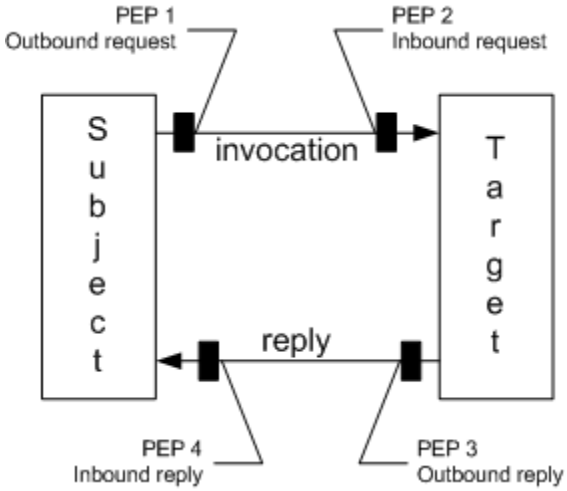

Figure 3-3. PAF framework in Ponder2 [88]

For current research purposes, it is possible to enforce authorisation policies on the EUD in two PEPs. One PEP prevents the EUD from harming the network, which occurs in the current provider AR as PEP1. Also, PEP4 is used to prevent the EUD from acquiring a service connection using the core-end point server to remove the user's access. Chapter 5 explains how this research project enforces these policies.

PAF supports two types of authorisation policy, negative and positive, which may conflict for the same subject. This issue and its solution are discussed in Section 3.7.

3.7.2 Self-managed cell

This subsection explains the SMC in more detail. As previously mentioned, the SMC manages a set of hardware and software components, such as those in large-scale network systems. These components interact via links or protocols in the network. The SMC contains a set of services that interact using an event bus, which supports the event-driven management system requirements [72]. The proposed management system is event-driven, and it uses an event bus to transmit events from one managed object to another. When an event is transmitted from an EUD in the Y-Comm network architecture, the security engine receives this event and makes a decision about what action should be taken. Additional details of the proposed system are explained in the following chapters.

The event bus in the SMC has been designed using Ponder2, which allowed the authors to achieve several goals. First, the SMC increases the dynamics of the system, because the sender of the event does not need to identify its recipient. Thus, the SMC does not disturb other managed objects [72]. For example, when a security system enforces a

negative authorisation policy based on a detected event, it does not need to disturb the other components of the system. Second, the event bus allows many components to respond to an event at the same time to achieve different goals, which is necessary in the case of the current research, because it allows the proposed security management system to respond to a malicious event using several actions. This research considers this design an advantage in achieving the security management system's goals. Finally, the event bus can be used to transmit data. However, this research does not consider this advantage, because there is no need to transfer application data in the event bus.

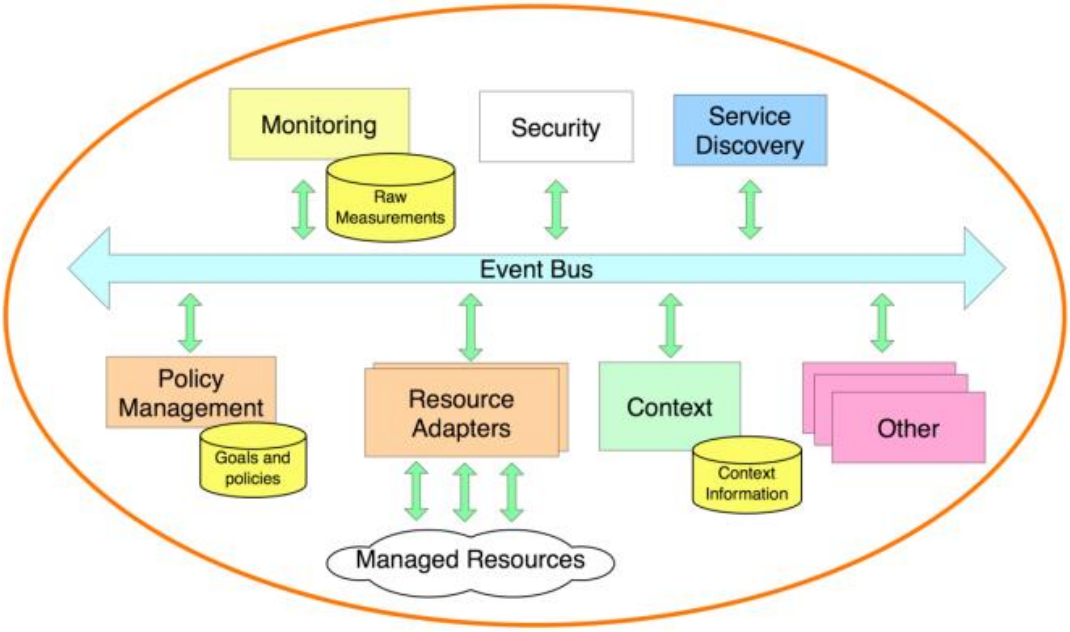

Figure 3-4. Self-managed cell architecture [89]

Figure 3.4 illustrates the SMC's architecture, showing how the services interact through the event bus. These services differ with the system that initiates the SMC. However, some components are present in all systems, such as the event bus, a discovery service, and a policy service [72, 90]. The proposed security management system has two main

components: the policy service and the event bus. The discovery service is not considered in this research, because every new mobile device that joins the network should use a network-level agreement and is allowed into the network by the core-end point CA3C server. Chapter 5 explains the components of the proposed security management system in more detail.

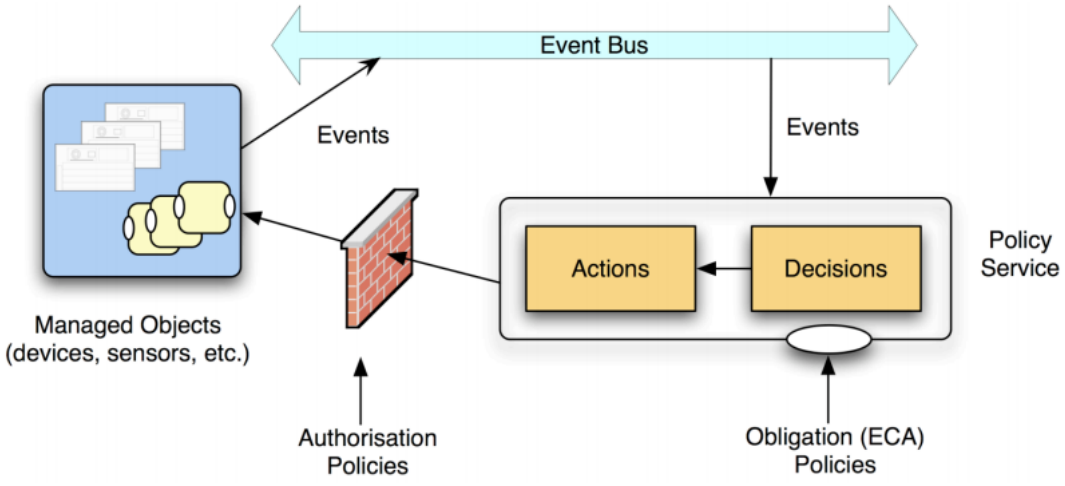

Figure 3-5. Policy-based feedback loop [72]

The SMC's main feature is its self-management, which is achieved by incorporating a basic feedback control loop. Figure 3.5 shows this loop and how the system moves from one state to another after an event occurs. Two types of policy are seen to work together to achieve the system's goals: an obligation policy is triggered by an event, then an authorisation policy completes the action required by the obligation policy. The events are driven by managed objects within the system, and the same or other managed objects enforce the policy [72, 89]. This loop has been employed in this research environment, and Chapter 5 explains the components that drive the events and the components' actions in more detail.

Policies in this basic loop can be enabled and disabled to meet different system goals, and these changes can be made without changing the codes for the managed objects [72]. The following are basic examples of how the security management system can control the behaviour of the components in the Y-Comm network:

```
on EndUserDevice(behaviour) do
    if behaviour == "malicious" then
        policies/normal.disable(); policies/active.enable()
```

This policy is triggered by malicious behaviour in the system and executes the appropriate policy. The malicious behaviour is detected by the IA in the EUD. As this sample code shows, the EndUserDevice is a managed object. The event is transmitted via the event bus and activates or disables a policy. The following policy denies access to the EUD using a negative authorisation policy:

```
auth- /EndUserDevice → /Ycomm.{denyAccess, stop}
```

The policy service in the event bus supports changes in policies without the need to change the codes of the components and without shutting down the system. However, an adapter object is needed for each component to perform an action on other components [72]. Chapter 5 explains how the components of the proposed security management system interact to perform management actions.

3.8 Resolution of policy conflict

This section reviews policy conflicts and how Ponder2 addresses these. Many research studies [91-93] have sought ways to detect and resolve such conflicts. This research uses the features of Ponder2 to address this common issue in policy-based systems and to resolve it when it occurs in the network. Policy conflicts arise in policy-based systems because the environments contain a large number of objects, leading to errors or conflicts in the administrators' requirements. Conflict occurs when there are two authorisation policies: one that permits the subject's activity and another that forbids it. When diverse management functions apply different policies to objects of the system, Ponder2 provides a strategy to resolve the conflict by dynamically determining which takes precedence: when conflict occurs between two policies, precedence is given to the more specific policy [94].

For example, when there is a policy *p1* that has a domain conflict with a policy *p2* for a subdomain, *p2* takes precedence, because *p2* applies to a subdomain and is thus more specific than *p1*, which applies to a whole domain. Figure 3-6 shows that *px* is applied to a subdomain and *pa* is applied to a domain. In other words, *px* is more specific than *pa*. In any conflict between the two, *px* takes precedence because it is more specific. The proposed security management system resolves this conflict using Ponder2's features. The security engine then enforces the appropriate policy.

This feature of Ponder2 is useful to the proposed SMS4HN and meets the security requirement of 4G heterogeneous networks. In addition, this conflict resolution strategy works at run-time, which makes it more dynamic.

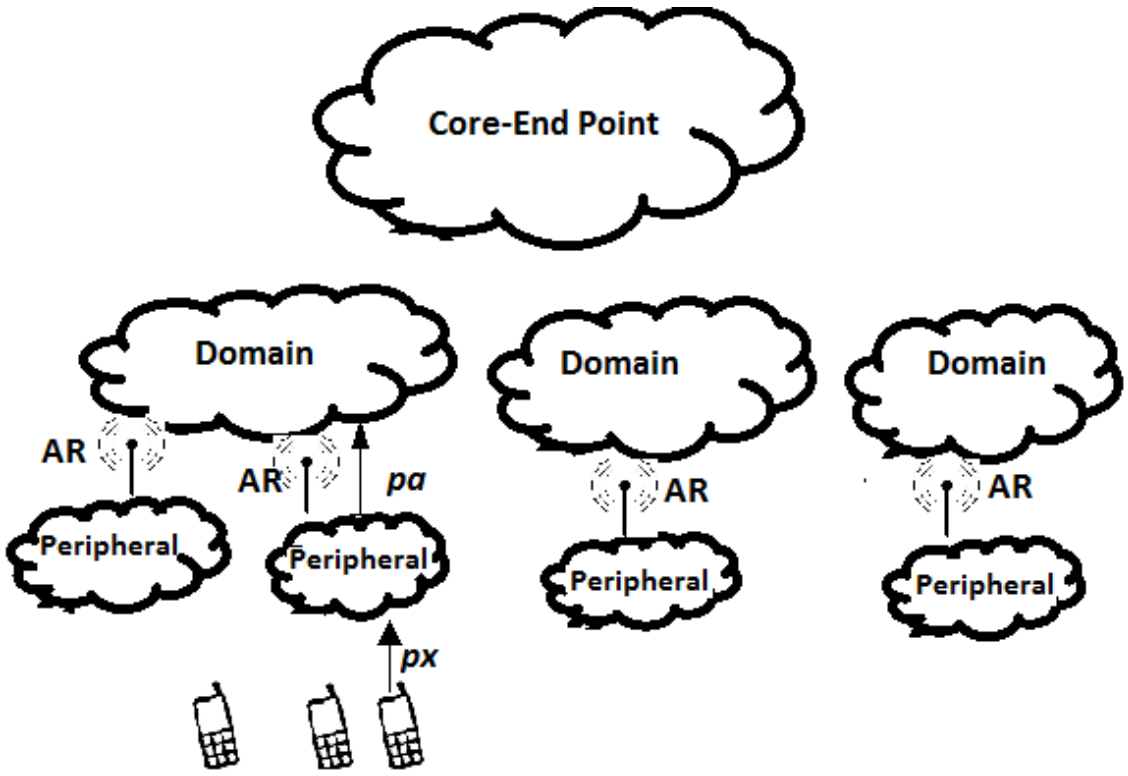

Figure 3-6. Policy conflict resolution in the Y-Comm network structure (adapted from [83])

3.9 Summary

This chapter has introduced the structure of the Y-Comm network, which is the environment for this research project, including the important levels of the network: the core network and the peripheral networks attached to it.

- The chapter has provided justification for this research project's assumption that access to users' information is more harmful in this type of network.
- It has discussed the security requirements of 4G mobile networks.
- It has explained the assumptions in the proposed security management system and built the system based on these assumptions.

- It has detailed the fundamentals of Ponder2, which is the policy specification language for this research project.
- It has also explained policy conflict, an important problem that occurs in most policy-based systems, and discussed a conflict resolution strategy.

The next chapter will explain the framework of the proposed security management system and each component in the system in detail. It will also discuss the design considerations of this research project.

Chapter 4: Security Management System Framework for 4G Heterogeneous Networks

Objectives:

- Present an overview of the SMS4HN framework;
- Explain the framework architecture;
- Explain how the framework components interact.

4.1 Introduction

Previous chapters have discussed the complicated research project environment, the Y-Comm network. The motivation for the proposed security management system and the reason for choosing a policy-based approach were also explained. This chapter presents an overview of the proposed SMS4HN framework and explains its architecture in detail. It also details how the components of the proposed framework interact with the Y-Comm network architecture. In previous chapters, it was noted that this architecture has heterogeneous components, imposing design considerations on this research project which are explained in this chapter. It is structured as follows: Section 4.2 defines the problem, Section 4.3 presents an overview of the framework and explains the components, then Section 4.4 explains the sequence of messages in the proposed security management system.

4.2 Problem definition

This section explains the Y-Comm network framework problem and why a security management system is needed. Because the Y-Comm network is a 4G heterogeneous network, it must meet all of the security requirement of 4G mobile networks, as discussed in Chapter 3. In particular, the security analysis of the Y-Comm network in Chapter 2 showed that one important requirement should be clearly met: a security management system is necessary in this environment to detect any security violations and handle these using a policy-based system.

Smartphones in the Y-Comm network environment have more network resource access privileges than in previous generations of mobile networks. Although this is considered an improvement to the service provided to users, an EUD may be attacked or used as an attack tool to harm the Y-Comm network. The security management system should be aware of this security concern and should propose an appropriate solution. However, the convergence of wired and wireless networks makes it difficult to propose a policy-based system that works within heterogeneous systems with different security levels. This research project considered this difficulty and chose to include tools that would suit the Y-Comm environment.

The research aims to meet the 4G mobile network security requirement which states that a security management system must protect end user devices from being used as attack tools. This requirement has not yet been met by the security models of the Y-Comm network; when an attacker controls an EUD, it can be used as an attack tool, due to users' increased privileges with network resources and applications. The increased smartness of phones has caused a concomitant increase in security threats, as explained in Chapter 2. This problem motivates this research to develop a solution that protects important future heterogeneous networks, such as the Y-Comm network.

4.3 Framework overview

This section describes the proposed security management system framework. A management layer is proposed at the top of the Y-Comm network architecture. This layer works as a security management system to detect predefined security violations in the network. It then contains these violations and prevents them from propagating and

harming the network as a whole. The security requirements of 4G networks, as explained in Chapter 3, state that the EUD should be protected from abuse and should prevent an attacked EUD from being used as an attack tool. This requirement has not been previously satisfied in the Y-Comm network architecture and needs to be addressed, as explained Chapters 2 and 3.

An attack on the network can occur when an EUD's user privileges in the network are stolen. User privileges are considered sensitive data, as explained in Chapter 3, so stealing this sensitive data triggers the security management system. To detect this type of security violation, an IA in the EUD is proposed. A full explanation of the IA functions is provided in section 4.4.2.1.

4.3.1 The management layer

The management layer, located above the security models in the Y-Comm network architecture, is a policy-based system that interacts with the main network components. Figure 4-1 shows the management layer, which is composed of the security management system; this is explained in the next section.

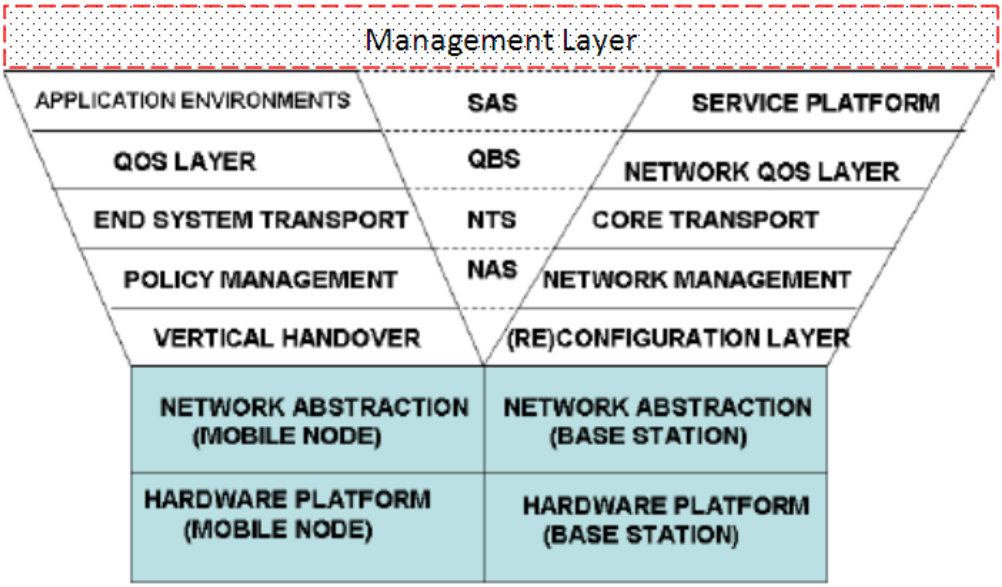

Figure 4-1. Management layer of the Y-Comm network framework (adapted from [36])

4.3.2 The framework

The main goal of the proposed security management framework is to detect an attack on an EUD and prevent it from being used as an attack tool. Thus, the main components of the system are an IA, a security engine, a security administrator, and a security database. This research proposes an architecture for these components. However, the prototype implementation applies only to the security engine, due to the limited time and resources available to implement such a large system in the difficult environment of the Y-Comm network. Figure 4-2 shows the architecture of the proposed security management system.

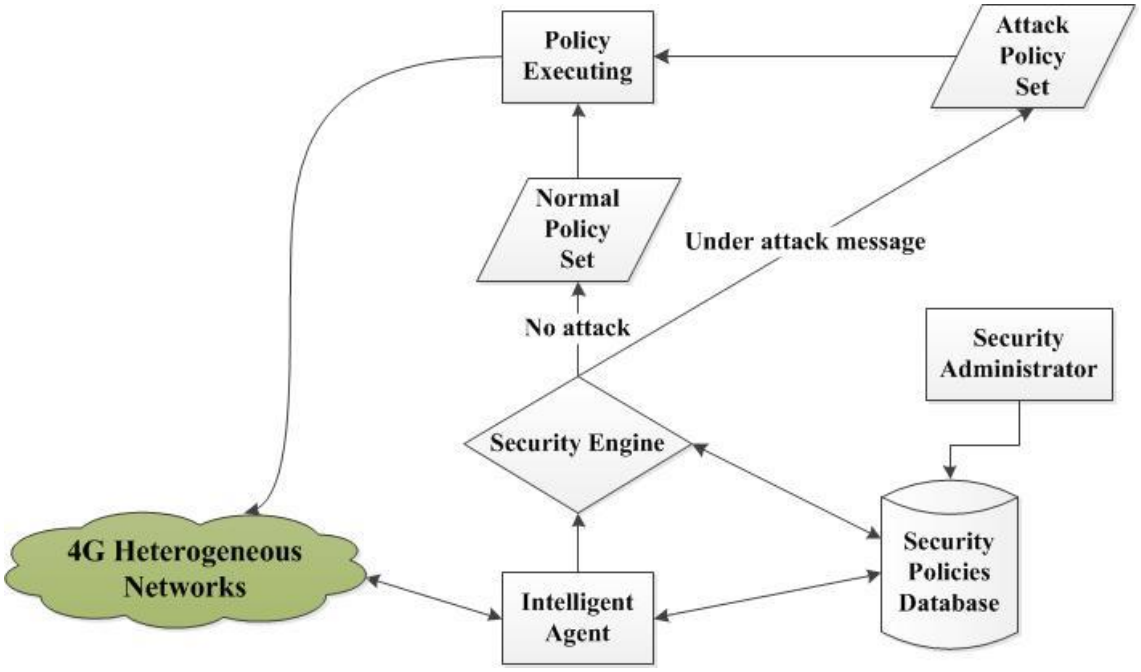

Figure 4-2. The proposed SMS4HN functional diagram

4.3.2.1 IA

The IA is a piece of software located in the EUD which works with the security engine and warns it when a security violation occurs. The IA was designed to follow the ITU-T recommendation M.3400. As explained in Chapter 2, the recommendation suggests a function set requiring the security management system to monitor internal users for theft of their service, because the purpose of such theft may be to target an EUD and use it to attack network resources. This function set meets an important security requirement of 4G heterogeneous networks explained in Chapter 3: to protect the network by preventing a legal EUD from being used as an attack tool.

The IA has four main functions: (1) it collects information from the EUD based on the security engine's management policies, (2) it analyses this information and checks for

malicious events, (3) it prepares a report and saves changes between previous and newly collected information, and (4) it sends the report to the security engine, so that it can make a decision and apply the appropriate policy. Figure 4-3 shows the four main functions of the IA.

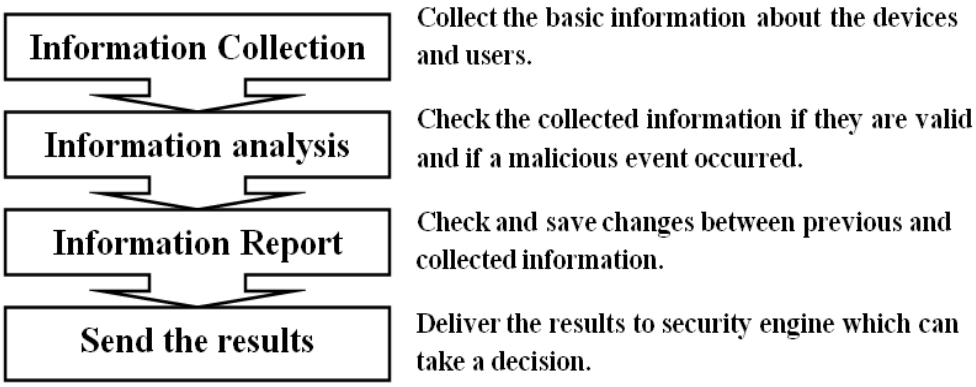

Figure 4-3. The IA's internal processes

4.3.2.2 The security engine

The security engine obtains information from the IA and makes decisions based on this information, which can then trigger the security engine to apply predefined security policies. The security engine identifies the appropriate policy based on many factors, including the type of attack, the type of EUD, possible vulnerabilities in the same node, and previous records in the database, then it stores this information in the security database for future use.

An important threat is a malicious attacker's attempt to access the configuration files to steal a user's privileges and attack the network. We believe that such attacks are able to harm the network. Why they are dangerous and how the system can be prepared for them was explained in Chapter 3. When the IA detects this kind of attack, it warns the

security engine, which isolates the EUD that is being used as an attack tool. The isolation of the EUD is based on the exception report action function set, which is part of the ITU-T recommendation explained in Chapter 2. The exception report action states that the security management system should limit security breaches using security tools, such as isolations. This action occurs when two main components of the Y-Comm network architecture, the CA3C server, which contains the network-level agreement (NLA), and the related access router, interact. When the security engine removes a user's access, that user cannot move or access another network provider within the network. However, the user is still connected to the current service provider, which is why the security engine needs to interact with the access router to deny access and isolate this malicious device.

4.3.2.3 The security administrator

The proposed architecture was designed to include a security administrator, which discovers inconsistencies in policy statements and specifies new policies when applicable. The security administrator can activate and deactivate policies via the shell interface, which is part of the Ponder2 framework and supports Ponder2 systems. The functionalities of the security administrator, such as discovering inconsistencies in policy statements, should be automated for many reasons. First, human network management capabilities are unable to match the rapid changes in the Y-Comm network's topology. Second, the security administrator needs to be automated to prevent malicious attacks after these changes. Finally, the limitations of static policies are well known, and dynamic policies are more efficient and responsive to change [78].

Automated services have not been implemented in the work reported in this thesis, but the functions of the security administrator are explained.

4.3.2.4 The security policies database

The security database (DB) is part of the SMS4HN framework. It is used to store the details of malicious events received from the IA. These details are kept for future analysis and improvements in the SMS4HN. The DB is accessed from the security engine, and this research study uses the output text file to store malicious event records. The next chapter shows how the SMS4HN stores the malicious event records it receives from the IA.

4.4 Sequence of messages in the SMS4HN

This section explains the sequence of messages and interactions between the components of the proposed system with the Y-Comm network architecture. The SMS4HN works during either of two cases: normal and dangerous. In a normal case, there is no security violation or malicious behaviour in the network (shown in Figure 4.4). The IA's analysis shows no malicious behaviour or security violation in the EUD, so the security engine does not need to take any action. Conversely, when a security violation occurs, the IA sends a report to the security engine, which records the event in the database and executes the appropriate policy, as shown in Figure 4.5.

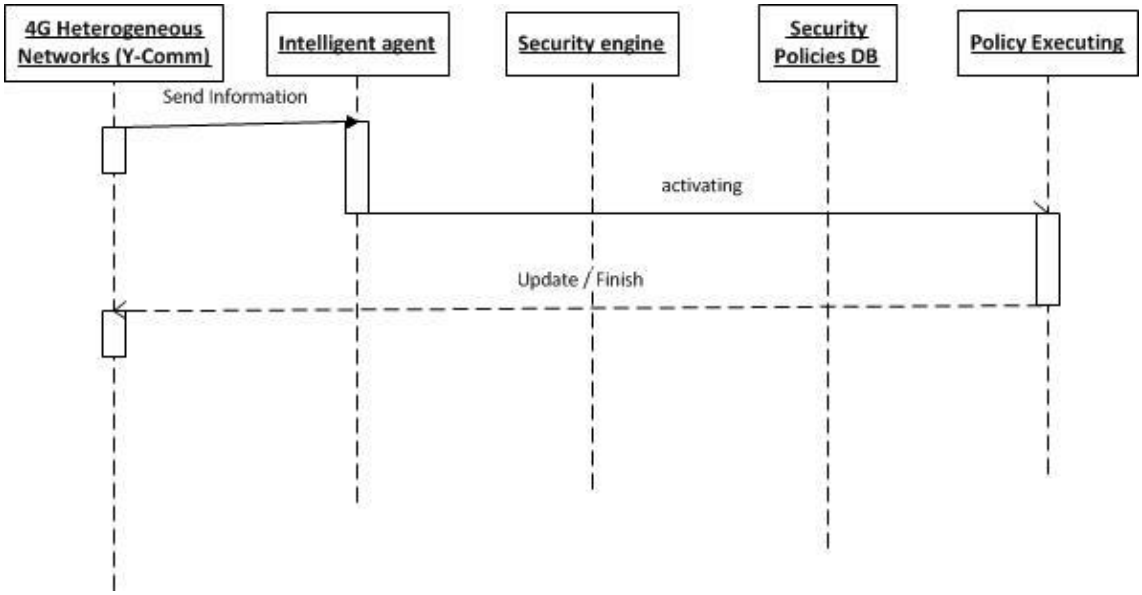

Figure 4-4. Sequence diagram of the SMS4HN when there are no security violations

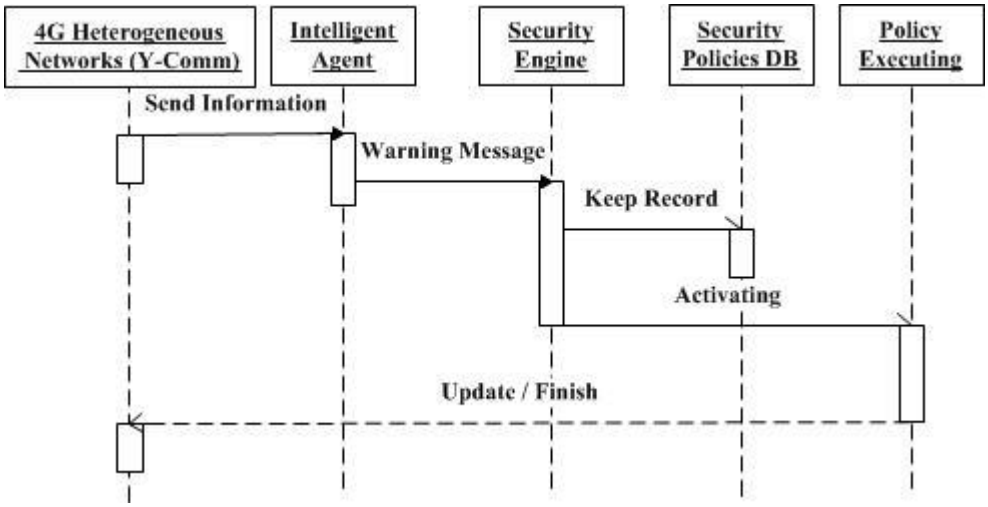

Figure 4-5. Sequence diagram of the SMS4HN during a security violation

4.5 **Summary**

- This chapter began by defining the problem that this research seeks to solve. It explained that the Y-Comm network framework needs a security management system that meets the security requirements of a 4G mobile network.
- It next outlined the proposed framework, giving details of each component. Design considerations were also discussed.
- Finally, the sequence of messages between the SMS4HN components was illustrated using sequence diagrams.

The next chapter explains how this system and its components are implemented, giving details of the interactions between the components of the Y-Comm network and those of the proposed framework.

Chapter 5: **Prototype Implementation of the SMS4HN**

Objectives:

- Describe how the proposed SMS4HN deals with a malicious event;
- Present the implementation specifications of the proposed SMS4HN;
- Explain the policy enforcement points in the proposed SMS4HN;
- Describe the SMS4HN's response to malicious events;
- Explain the problems facing the SMS4HN in the Y-Comm network environment.

5.1 Introduction

The previous chapter restated the research problem and proposed the SMS4HN for the Y-Comm network architecture. It explained the importance of using a policy-based approach to meet the security requirements of 4G mobile networks. This chapter describes the implementation of the prototype security engine, which is the brain of the SMS4HN. It also explains how the SMS4HN works in this heterogeneous environment, which is known for its difficulty and complexity, as discussed in Chapter 2. The other system components, such as the IA, can be implemented in future work in this field. The present research project designs the SMS4HN components and the Y-Comm network as managed objects to improve the interactivity and interoperability of the SMS4HN components. This research focuses on addressing the challenges of a policy-based system capable of protecting Y-Comm networks. It offers solutions to the challenges of the Y-Comm network environment, so that this security management system can be proposed. This chapter discusses the detailed implementation of the SMS4HN and shows how the its components interact with those of the Y-Comm network.

One of the challenges facing this research has been the identification of the most appropriate of the many policy specification languages available to implement the proposed SMS4HN, given the complexity and the difficulty of the Y-Comm network environment. This language chosen is Ponder2, because it is flexible and suitable for environments containing diverse technologies, such as optical and 3G networks. Ponder2 also adopts the concept of a self-managed cell, which supports dynamic policy

management for large-scale systems. Chapter 2 provides a detailed explanation of the features of Ponder2 and the justification for choosing it.

This chapter reports implementation of the prototype security engine using Ponder2. Section 5.2 explores how the SMS4HN interacts with the Y-Comm network components during a malicious event. Section 5.3 presents details of the SMS4HN specification, including the obligation and authorisation policy commands, explains the mechanism used to solve conflict among Y-Comm network authorisation policies, and explains communication between managed objects. Section 5.4 describes the policy enforcement points that are used in the proposed policy-based system. Section 5.5 explains the working of the policy feedback loop, showing the interaction between the components of the SMS4HN and the Y-Comm network, and Section 5.6 explains the SMS4HN's response to Y-Comm network events.

5.2 Response of SMS4HN to a malicious event

This section illustrates how the SMS4HN components interact during a malicious event. The chapter begins with this scenario to provide an overview of the SMS4HN process and its interaction with Y-Comm. The scenario assumes that the EUD contains an IA, so is able to interact with the security engine. The SMS4HN currently contains an events generator to trigger the security engine, but this scenario assumes that there is an EUD connected to the 4G heterogeneous network (Y-Comm). This EUD has the privileges necessary to access network applications and resources, including the privilege to access and modify applications on a peripheral network in the Y-Comm

network. These privileges are considered sensitive data and are stored as configuration files, such as password files.

In this scenario, an attacker is trying to steal these privileges to prepare an attack on the whole Y-Comm network. In other words, the EUD has become an attack tool in the hands of the attacker, who has taken the EUD's identity for the purpose of conducting activities in the user's name, such as voting or accessing the user's financial accounts [84]. There is a strong possibility that this kind of attack will happen to the Y-Comm network, and that it will be more dangerous, due to the additional network privileges that the user has on the Y-Comm network, as explained in Chapter 2. The openness and heterogeneousness of the network makes it more vulnerable to this type of attack.

The EUD contains an IA to detect malicious behaviour, and this sends the security engine a message containing a mobile equipment identifier (MEID), which is a unique number [95]. The MEID was proposed by the 3rd Generation Partner Project to provide each mobile device with a new unique global identifier that is stored in it. The IA also sends details of the attack type, date, and time to the security engine, which then checks to determine if the event meets the conditions necessary for it to be considered malicious. Next, the security engine takes action, enforcing the appropriate policies, which in this case involves removing the user's access by modifying the NLA in the CA3C of the core network. It also enforces the current AR, which gives the EUD the connection to stop providing the service. The enforcement of policies occurs at two policy enforcement points, the AR and the CA3C. Once the policies have been enforced, the security engine keeps a record in the database for future analysis and for the development of new policies.

This scenario is presented here to demonstrate how this research has implemented the SMS4HN so that it works with the Y-Comm network. The following sections detail the specifications of the proposed SMS4HN.

5.3 SMS4HN specifications

This section discusses the proposed SMS4HN specification details. The approach of this security management system is policy-based, which means that the system acts on obligation policies and authorisation policies. Obligation policies are specified in the ECA format, so they respond to security violation events. When an event occurs and the condition is true, an appropriate policy is applied.

This research project creates managed objects for the required Y-Comm network components so that they interoperate within the proposed system. These managed objects, including the EUD, DB, AR, NLA, and the warning managed object, then interoperate to achieve the goals of the security management system.

5.3.1 SMS4HN obligation policies

This subsection explains how this research project identified the SMS4HN obligation policies. Figure 5-1 shows a piece of code that explains how to create an ECA policy and how the SMS4HN interacts with the managed objects. In line 1, the system creates an ECA policy to check whether there is a security violation detected by the IA and received as an event. The system receives the event in line 2, which contains the four attributes: EUD ID, attack type, attack date, and attack time. The condition in Ponder2 is expressed as those in lines 3 and 4. The SMS4HN checks whether the condition is

satisfied, then it activates the policy. When the ECA policy is activated, the actions comprise four main steps. First, the system keeps a record in the database and sends the four attributes to the DB managed object. Second, it interacts with the NLA managed object and executes the function that removes the user's access. It also sends an EUD ID to the NLA managed object, as shows in line 7. Third, the system interacts with the AR managed object to activate the stopAccess function, which means that it stops providing service to this EUD. It also sends the EUD ID. Finally, the system warns the system administrator through a warning managed object, as shows in line 9.

```
// Create a new policy to prevent the EUD from accessing the network in case of attack
1.  policy := root/factory/ecapolicy create.
2.  policy event: root/event/eudValue.
3.  policy condition: [ :eudID :AttackType :Date :Time |
4.          root print: "Checking End User Device: "+ eudID + " with attack type: " +
    AttackType.
                AttackType == "ConfigAccess")].

5.  policy action: [
6.          root/DB keepRecord: eudID, AttackType, Date, Time.
7.          root/NLA removeAccess: eudID .
8.          root/AccessRouter stopAccess: eudID.
9.          root/warning setWarning: true; show ].
10.     root/policy at: "ConfigAccess" put: policy.
11.     policy active: true.
```

Figure 5-1. ECA policy in SMS4HN.

Another piece of code is shown in Figure 5-2, representing the creation of an event that is received from the EUD managed object. The system loads the event in the event bus, which transmits events between the managed objects, as explained in Chapter 3, so it interoperates with the ECA policy. The SMS4HN first creates a template for the event, then it defines the attributes of the event, as in line 1. This template is now ready to load, as illustrated in Figure 5-2. This step is necessary because it allows the ECA

policy to invoke the attributes of the events and check their values when an event occurs.

```
// Create a malicious event type for the system utility to send. This event type has
// four attributes (end user device ID, attack type , attack date , attack time).
1. template := root/factory/event create: #( "eudID" "attackType" "attackDate"
   "attackTime" ).
2. root/event at: "maliciousEvent" put: template.
```

Figure 5-2. Event template of SMS4HN.

These pieces of code are presented to show how the system employs the features of Ponder2 to deal with the Y-Comm network components. Additional details concerning the proposed SMS4HN codes are presented in the appendix.

5.3.2 Authorisation policies in SMS4HN

This subsection explores how the proposed SMS4HN uses authorisation policies to complete the security management tasks. As mentioned in Chapter 2, authorisation policies may be either positive or negative. This research uses negative authorisation as part of the SMC policy feedback loop, which has two parts: the obligation policy, triggered by an event that happens in the EUD, and an action specified by this proposed policy-based system, which removes the user's access, as explained in Section 5.2. In other words, a negative authorisation policy should be specified on the EUD. Figure 5-3

lists the commands required to specify a negative authorisation policy in the proposed SMS4HN.

```
newauthpol := root/factory/authpolicy.
root/authdom at: “authEUD” put:
                                    ( newauthpol
                                    subject: root/domainA/EUD
                                    action: “tcp”
                                    target: root/domain/AR
                                    focus: “t”).
root/authdom/authEUD reqneg.
root/authdom/authEUD repneg.
root/authdom/authEUD active: true.
```

Figure 5-3. The authorisation policy commands in SMS4HN.

The commands in Figure 5-3 first create an authorisation policy, then specify the subject, which is the EUD. Next, they specify the target, which is the AR. The last three commands are to refuse a request received from the EUD, not to reply to this EUD if it sends a joining request message, and to activate this policy.

```
root/authdom/authEUD active: true.
```

The activation command, repeated above, can be overridden by another command to deactivate the policy if the security engine requires this.

5.3.2.1 Conflict between Y-Comm network authorisation policies

This subsection explains how the proposed SMS4HN addresses authorisation policy conflicts, a common problem in policy-based systems, as stated in Chapter 3. While the conflict resolution strategy in Ponder2 is outlined in Chapter 3, this subsection explains this solution in more detail. It also gives the example of a scenario expected in the Y-Comm network environment, to clarify how the SMS4HN works.

Conflict between Y-Comm network authorisation policies can occur if a peripheral network gives all end users access to network resources, then the security engine enforces a negative authorisation policy on an EUD because the IA has detected a malicious event. Figure 5-4 illustrates such an abstract conflict.

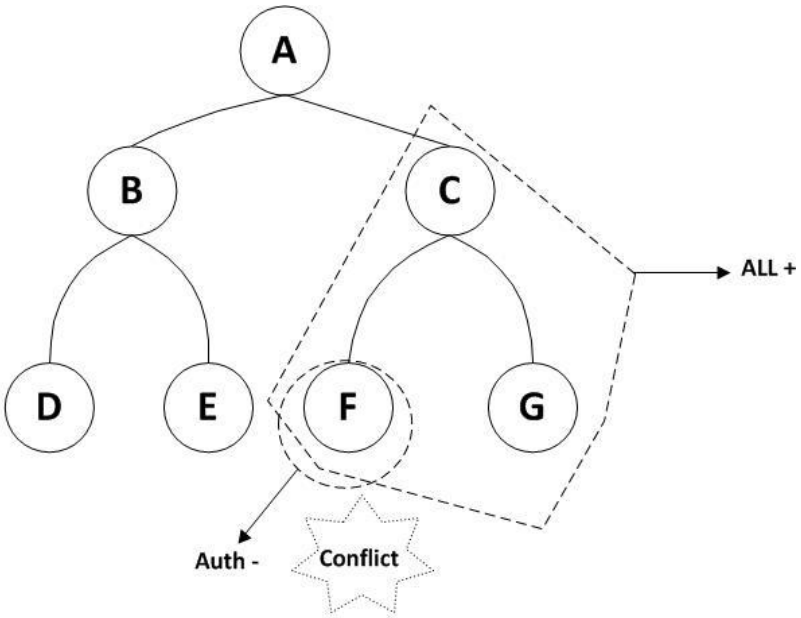

Figure 5-4. Example of an authorisation policy conflict.

Figure 5-4 shows that domain *C* (a peripheral network) gives a positive authorisation to all of its nodes. However, node *F* (an EUD) has a negative SMS4HN authorisation

policy, which conflicts with the domain-wide positive authorisation. A conflict resolution policy is then applied, giving precedence to the policy that is more specific. In this case, the negative authorisation applied to the specific EUD (node *F*) will take precedence over the policy specified in domain *C*. However, this precedence is not always given to more specific policies, and a global policy may override a specific policy in some situations. Therefore, the command *final* in Ponder2 guarantees that the special authorisation policy will not be overridden.

Another conflict occurs when two opposing authorisation policies occur simultaneously, such as when both a positive and negative authorisation policy apply to a node *F* at the same time. Figure 5.5 illustrates such a conflict. In this case, the conflict resolution strategy states that the negative authorisation policy takes precedence over the positive one.

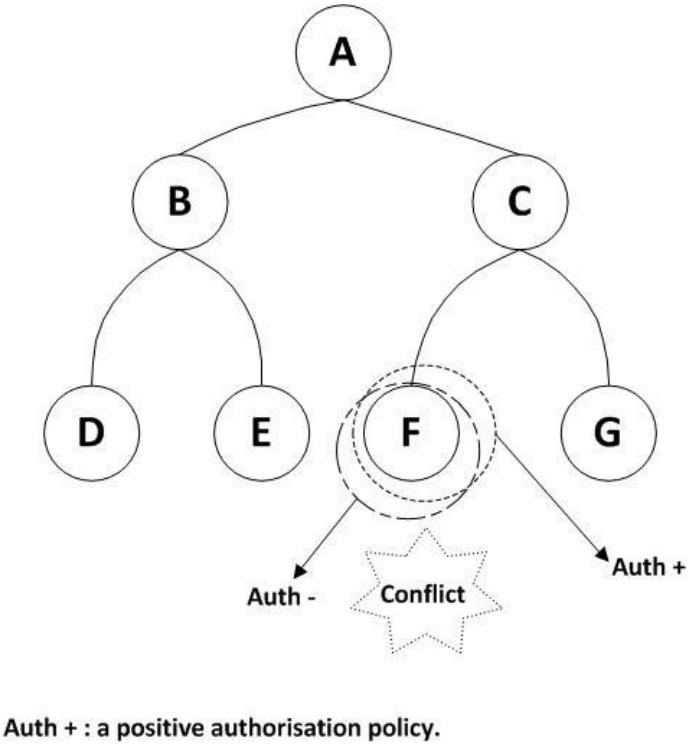

Figure 5-5. The conflict between two opposite authorisation policies.

The conflict resolution strategy in Ponder2 fits the present security management requirements, due to the design considerations of this language. The features of Ponder2 are thus suitable for the Y-Comm network environment.

5.3.3 Communication between managed objects in the SMS4HN

This subsection explains how the managed objects in this research project communicate with each other. They can receive PonderTalk commands via Remote Method Invocation (RMI), which specifies how Java objects are accessed to accomplish tasks on remote or local machines [96].

To send PonderTalk commands via RMI, the proposed SMS4HN needs to start PonderTalk inside the SMC. This is done by executing the following two command lines:

```
factory := root load: “PonderTalk”.
ptalk := factory create: “P2RMI”.
```

SMS4HN creates a malicious event template for the EUD, as shown in Figure 5-2. As the EUD is represented in this research project as a managed object, the following codes are used to send this event to the security engine:

```
template := root/factory/event create: #( "eudID" "attackType" "attackDate"
"attackTime" ).
root/event at: "maliciousEvent" put: template.
```

This EUD uses RMI to send the event to the security engine through the following lines of code:

```
java –classpath ponder2.jar net.ponder2.PonderTalk SMSRMI
‘ root/event/maliciousEvent #("eudID" "ConfigAccess" "23/05/2013" "20:31") . ’
```

The first line aims to contact the security engine using the SMSRMI’s address, then the second line sends the values and a message about this event to the security engine. The status of this message will be printed in an output file. If there is an error while the message is being sent, a warning message will be sent to the security administrator.

Another way to send messages between managed objects is via a web service. Although this research project does not currently use this method to send messages between managed objects in the Y-Comm network, web services can be used in future stages of the proposed integration of the SMS4HN into the network. The use of a web service may be made necessary by the technologies and large scale of the Y-Comm network. The proposed system can be easily modified to support communication using web services. Sending messages between managed objects in this way is clearly explained in [6]. The appendix to this thesis also contains the code required to send an event from the EUD managed object to the security engine using a web service.

The Java RMI registry must be running on each machine using Ponder2 to start communication between managed objects. Thus, this research assumes that each managed object represented is running an RMI registry.

The proposed SMS4HN interacts with the managed objects as if they were local. Communication between managed objects is transparent in PonderTalk, which requires only two things to send messages to them: a communication medium module and a unique reference identifier. The former is the ponder2comms module, and a unique reference identifier is assigned to each managed object by Ponder2.

5.3.4 Dealing with SMS4HN exceptions

This subsection explains how this research project deals with SMS4HN exceptions. Exceptions in Ponder2 are designed to address the errors that may occur while running a Ponder2 system. This research uses three main Ponder2 exception objects. First, the *Ponder2ArgumentException* is used if there is something wrong with the argument,

such as instances when an argument is missing from an event sent from the EUD to the security engine. This exception is also included in case of conversion errors that may occur between argument types, such as instances when a string type is expected and received as an integer. Second, the *Ponder2OperationException* is used to address errors in methods or operations. Finally, the *Ponder2RemoteException* is included to address communication errors between managed objects. If one of these exceptions is thrown, details will be added to the exception, such as the file name and line number.

5.3.5 Sending messages to managed objects in the SMS4HN

This section explains how the proposed SMS4HN sends PonderTalk messages to managed objects, such the AR and the EUD. This is done via the *operation* method call, as follows:

```
P2Object operation(P2Object EUD, String operation, P2Object args) throws Ponder2Exception;
```

The first argument in this operation method call identifies the source of the message. In our case, it is the EUD that generates the message and sends it to the security engine. The second argument is the PonderTalk message, such as `at: "eudValue" put: event`. The third argument in the operation call is the array of P2Objects that contain the values of the message arguments, such as the EUD ID, attack type, attack date, and attack time.

P2Object is the base class for the managed objects in Ponder2. It has been used in this research project to create the managed objects, each of which is sent through a PonderTalk message as a P2Object.

In Ponder2, managed objects do not need to know the protocols required to pass messages; they need to know only the class path. Ponder2comms.jar is a module in Ponder2 containing protocols that can send messages to and from managed objects. It is loaded when the Ponder2 system is started. Thus, when a managed object such as an AR sends a message to the security engine, the ponder2comms module will search for, load, and use the appropriate protocol module.

Although only two protocols (RMI and web service) are used in ponder2comms, it can support other protocols when the protocol module is written and included in the Ponder2 class path. Then the protocol module will be loaded when it is required, without the need to change the configuration settings in Ponder2. This ability to send messages between managed objects using other protocols is considered an advantage of Ponder2. Using other protocols is not necessary at this stage of this PhD project, but it may be needed in future stages of the proposed SMS4HN's deployment. The Y-Comm network is expected to contain different network technologies that may require other protocol modules to communicate. Lupu et al. used other protocols, such as IEEE 802.15.4, in their research study [72]. Thus, Ponder2 is able to support other protocols. The appendix to this thesis explains how to add a protocol for communication between managed objects.

5.4 SMS4HN policy enforcement points

As stated in Chapter 3, the authorisation framework in Ponder2 supports two authorisation enforcement points. Figure 5-6 shows two policy enforcement points that are enforced in the Y-Comm network architecture. The SMS4HN enforces PEP1 at the

core-end point, specifically in CA3C, which contains network level agreements. As mentioned in Chapter 3, these NLAs hold the terms of the user's access to the network services. Therefore, the SMS4HN interacts as a managed object that uses an NLA to enforce the policy to remove the user's access to the Y-Comm network. This policy enforcement occurs when security violation is detected that may harm the entire Y-Comm network, as explained in Chapter 4. However, when the user's access to CA3C is removed, the EUD is still connected to the peripheral network, which is why we need another PEP.

The second PEP is in the access router (AR) and stops the provision of the connection to the EUD. The enforced authorisation policies are negative, preventing the subject (the EUD) from accessing the target (the Y-Comm network).

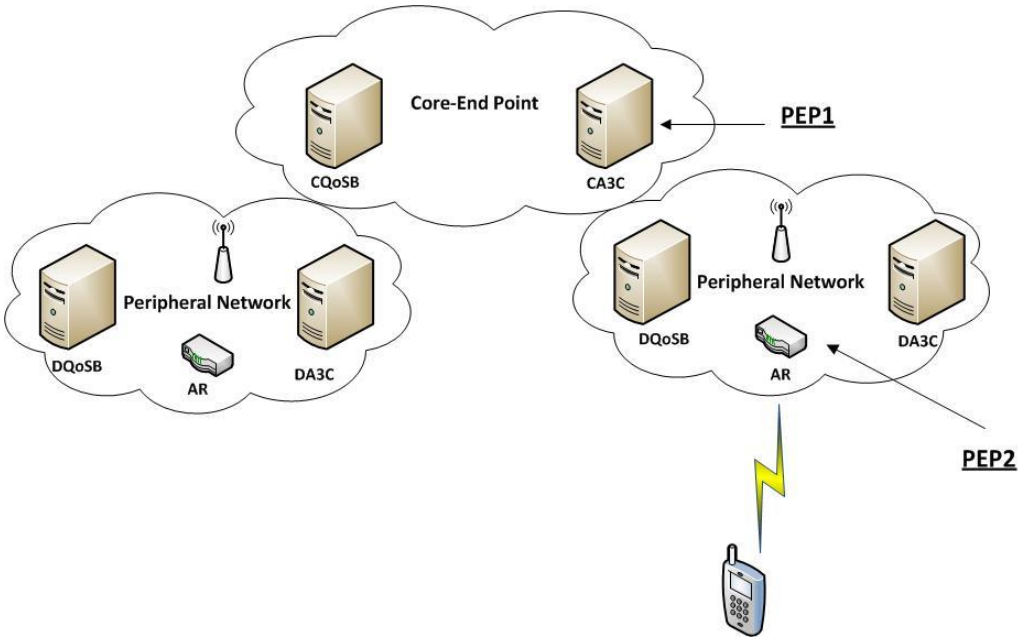

Figure 5-6. Policy enforcement points in SMS4HN (adapted from [83]).

In this project, PEPs are transparent to the security engine; they are wrapped with a piece of code which controls the messages that leave or are transferred to the managed object. For instance, the ARs in the Y-Comm network are managed objects and use various technologies. The use of wrapping code makes controlling the ARs easier. In other words, the security engine does not need to know the details of an AR to enforce the policies. The AR is one of the policy enforcement points in this research project, and the wrapping code can be used with all managed objects. This research assumes that the wrapping code will be implemented for the managed objects when the proposed SMS4HN is integrated into the Y-Comm network in the near future.

The wrapping code can be implemented using the Java Management Extension (JMX). JMX was used to wrap the managed objects in [94] because it is a powerful and efficient tool for managed objects using various technologies. Thus, the wrapping code for the ARs and the other managed objects should be the focus of future work.

5.5 The policy feedback loop in the SMS4HN

This section explains the use of the policy feedback loop as a part of a self-managed cell in this research project. The SMC with the policy feedback loop is a set of hardware and software components that receive an event, then deal with it via a policy. The background of the SMC and policy feedback loop is explained in Chapter 3. We also mentioned in Chapter 3 that the Y-Comm network components represented in this research project are managed objects which can be managed and controlled through a Ponder2 system.

A managed object, such as an EUD, generates events, which trigger the proposed SMS4HN. The SMS4HN analyses these events and applies the appropriate ECA obligation policy. An extension of the obligation policy is the authorisation policy, which is forced back on the components of the Y-Comm network. This interaction loop between the Y-Comm network components and the SMS4HN is called a policy feedback loop.

Figure 5-7 shows the SMS4HN policy feedback loop, which consists of the loop of interactions between the Y-Comm network components and the SMS4HN. The first managed object in this figure, the EUD, generates an event, which is transmitted to the security engine through the event bus. When the security engine receives the event, it determines what action should be taken, depending on the attack type; if the event is malicious, it takes action based on the obligation policy. The second half of the loop, after the action is taken, enforces the authorisation policies on the managed objects, such as the AR and the CA3C, in the Y-Comm network. The authorisation policies enforce two PEPs on two managed objects: the AR and the CA3C. This loop represents the SMC in the SMS4HN.

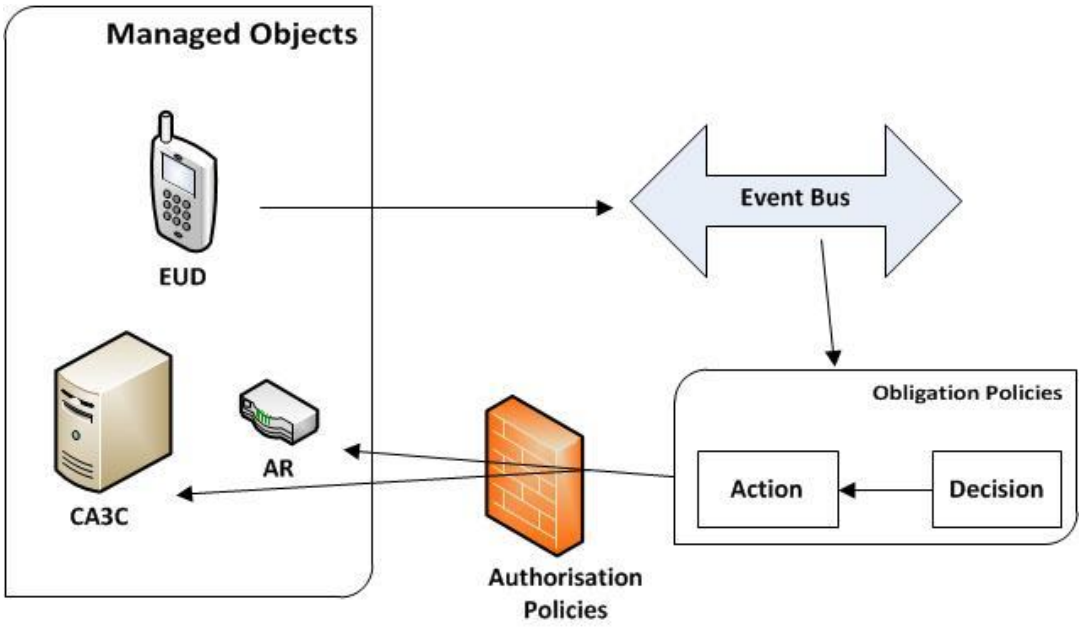

Figure 5-7. SMS4HN policy feedback loop

5.6 The SMS4HN's response to events

This section describes the SMS4HN's response to events in the Y-Comm network. This research project created an event generator to simulate an EUD in order to test the security engine's response to events. The system responds to malicious events generated by the event generator and enforces the policy. The SMS4HN interoperates these managed objects to achieve the security requirements of the Y-Comm network, such as preventing the EUD from being an attack tool. Figure 5-8 is a snapshot of such a response.

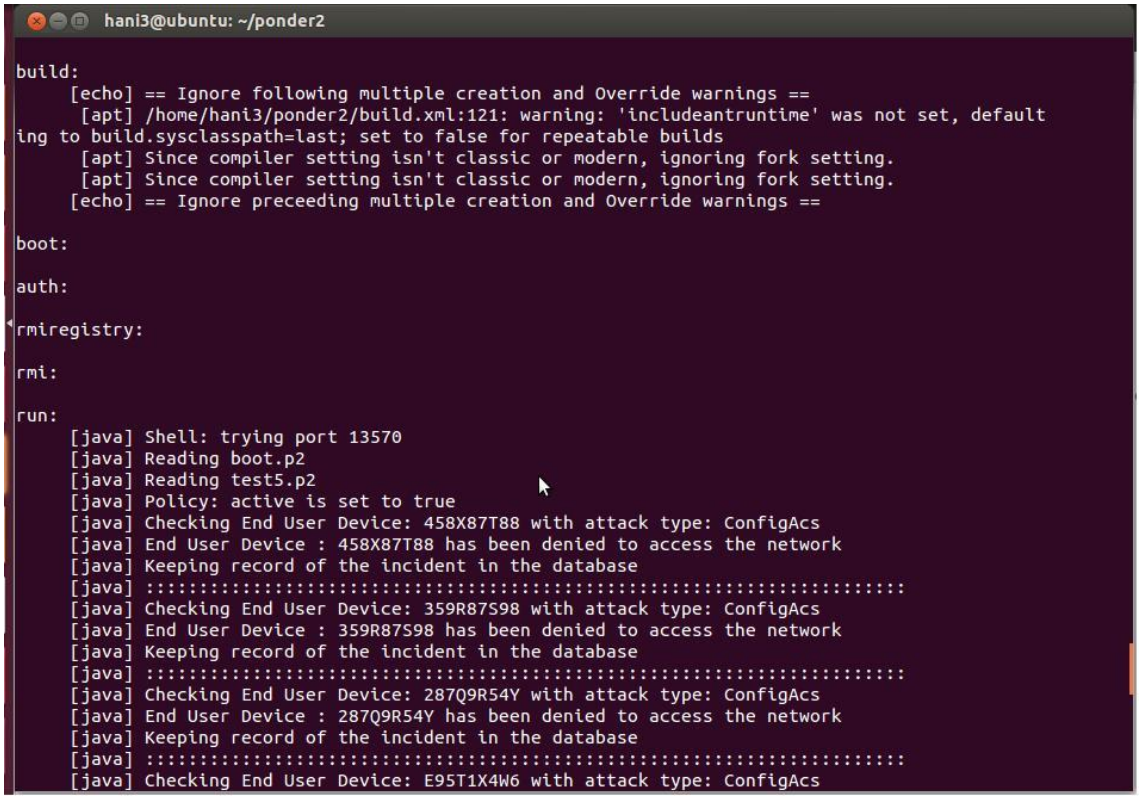

Figure 5-8. Snapshot of SMS4HN after it detects a malicious event.

Figure 5-8 shows the main steps that the system takes when it detects a malicious event. It activates the policy to deny the user's access and keeps a record with full details of the events, so that they can be analysed and used to extend the system in future work. The system then creates an output file to store the events that have occurred in the network. Figure 5-9 shows the output file of the stored malicious event record. This file, listing the EUD ID, attack type, attack date, and attack time, is needed for future improvement and for the analysis of the SMS4HN.

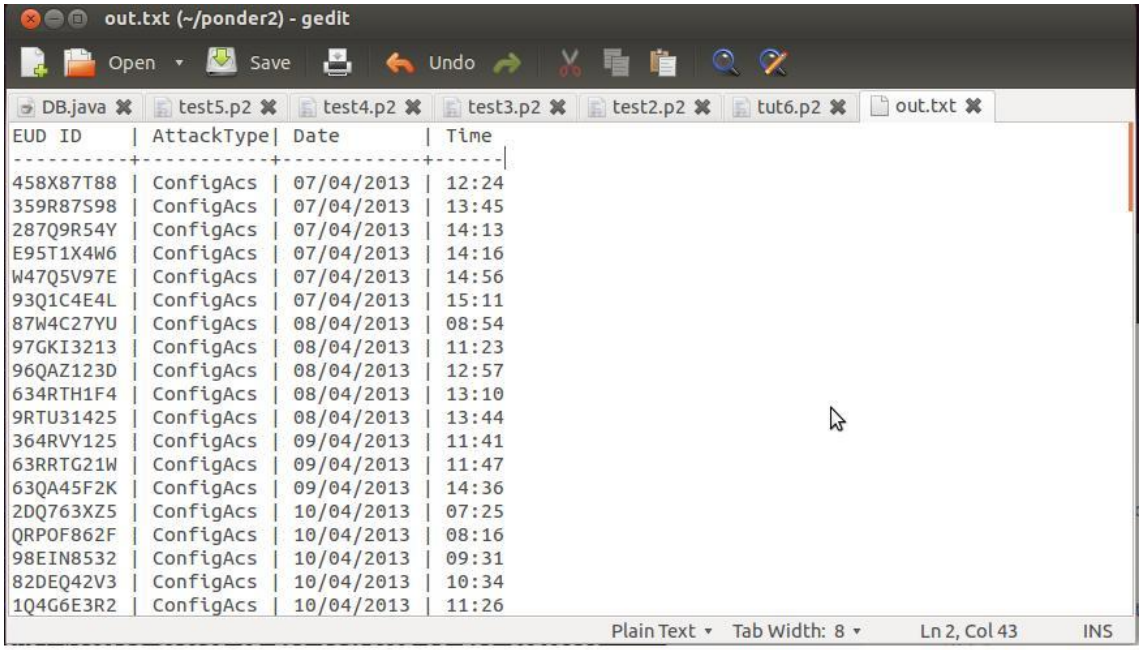

Figure 5-9. Snapshot of the output file of the malicious events record.

5.7 Summary

This chapter has detailed the implementation of the proposed SMS4HN.

- It began by explaining how the SMS4HN interacts with the Y-Comm network components in the case of a malicious event.
- Next, it set out the implementation specifications of the proposed SMS4HN. It explained how this research project has created managed objects and how they interoperate to achieve the project's goals.
- It then explained how this research implements obligation and authorisation policies, followed by an explanation of the mechanism to solve policy conflicts.

- It next explored communication among the managed objects and how the SMS4HN addresses exceptions and errors. The policy enforcement points in the Y-Comm network architecture used by the proposed SMS4HN were then explained.
- The chapter ended by outlining the response to events in the Y-Comm network.

The next chapter presents a case study for this research project, details the IA mechanisms, and gives an example of the Y-Comm network.

Chapter 6: **Case Study**

Objectives:

- Present a detailed scenario of the detection of a malicious event;
- Explain the vertical handover of EUDs in a Y-Comm network, including a case study;
- Give details of the SMS4HN processes in the Y-Comm case study;
- Explain the IA detection mechanism.

6.1 Introduction

The previous chapter explained SMS4HN implementation in detail. It also discussed the way in which this research project employs Ponder2 features when the SMS4HN is implemented in a Y-Comm network environment. This chapter presents a case study to establish the practical applicability of the proposed SMS4HN, giving details of the detection of a malicious event in an EUD. This case study uses the Android OS 4.0 (Ice Cream Sandwich) to detect a malicious event, because Android is a popular, open source operating system, which received 79% of malware attacks in 2013 [97]. The Android application store is an open market, which means that developers are able to upload their applications without filtration from any Android market authority. Although this is considered an advantage because it increases the number of Android applications available, these may call third party applications and encourage EUD misuse. These characteristics motivated the use of the Android OS for the proposed IA detection mechanism in this research. The IA design hypothesis is also stated, but the implementation and evaluation of the IA will be done in a future phase of this research project.

This chapter also shows how an EUD joins another peripheral network in the Y-Comm network environment, resulting in a vertical handover. Vertical handover is explained in this chapter because it proves the applicability of the SMS4HN to the Y-Comm network, which is explained using a case study of three existing networks in Britain.

Section 6.2 presents the case study scenario, which is explained further in later sections. Section 6.3 discusses vertical handover in the Y-Comm network using existing

networks. Section 6.4 shows how the proposed SMS4HN works in the Y-Comm network case study. Section 6.5 explains the malicious event detection mechanism in the EUD.

6.2 Case study scenario

This section presents a case study of a Y-Comm network composed of three existing networks. The first is the core-end point, in this case the Virgin Fibre Optic (VFO) network, which provides high-speed connections and manages peripheral wireless networks. The administrative authority for this heterogeneous environment is part of the ITU-T recommendation, as explained in Chapter 3. The other peripheral networks attached to the core network are the EE4G and Urban WiMax networks. Urban WiMax is an internet company that established the WiMax network in 2005. This network works in Central London and provides high-speed connections up to 1 Gbps [98]. EE is a digital communication company that provides a mobile communication network and 4G superfast connections for mobile users in Britain. These two networks have different network technologies, which the Y-Comm network combines: EE4G uses LTE technology, whereas the Urban network uses WiMax technology.

This research assumes that there is an EUD that operates the Android OS and is connected to the Y-Comm network. In the case study scenario, the EUD has been attacked in order to launch an attack on the whole network. This kind of attack has occurred in GSM networks, and there is a possibility that it may occur in the Y-Comm network, as explained in Chapter 5. This assumption is based on the increase in the number of privileges granted to EUDs, giving them the authority to access, modify, and

manage Urban network resources. Therefore, when these privileges are stolen or attacked to target the whole network, the EUD involved must be isolated, to prevent it from being used as an attack tool. The prevention and isolation of this EUD is part of the ITU-T M.3400 recommendation, as explained in Chapter 2.

The IA that is part of the proposed SMS4HN is responsible for detecting any malicious behaviour. However, many mechanisms and techniques can be used to detect malicious EUD behaviour, so the next section of this research presents an appropriate mechanism. Once the IA has detected malicious behaviour, it sends a message to the security engine, which applies the appropriate ECA obligation policy, then enforces the appropriate authorisation policy, as explained in detail in Section 6.4.

6.3 Vertical handover in the Y-Comm network

This section gives details of a vertical handover in the Y-Comm network through a case study that shows how this handover works in the real world.

Figure 6-1 shows the Y-Comm network used in the case study. The core network is the VFO network, containing the CA3C and CQoSB. The CA3C, as explained in Chapter 3, is responsible for authentication, authorisation, accounting, and cost, while the CQoSB is responsible for quality of service issues. In this case study, there are two wireless technologies (peripheral networks) connected to the VFO network: Urban WiMax and EE4G, which are official working technologies in Britain.

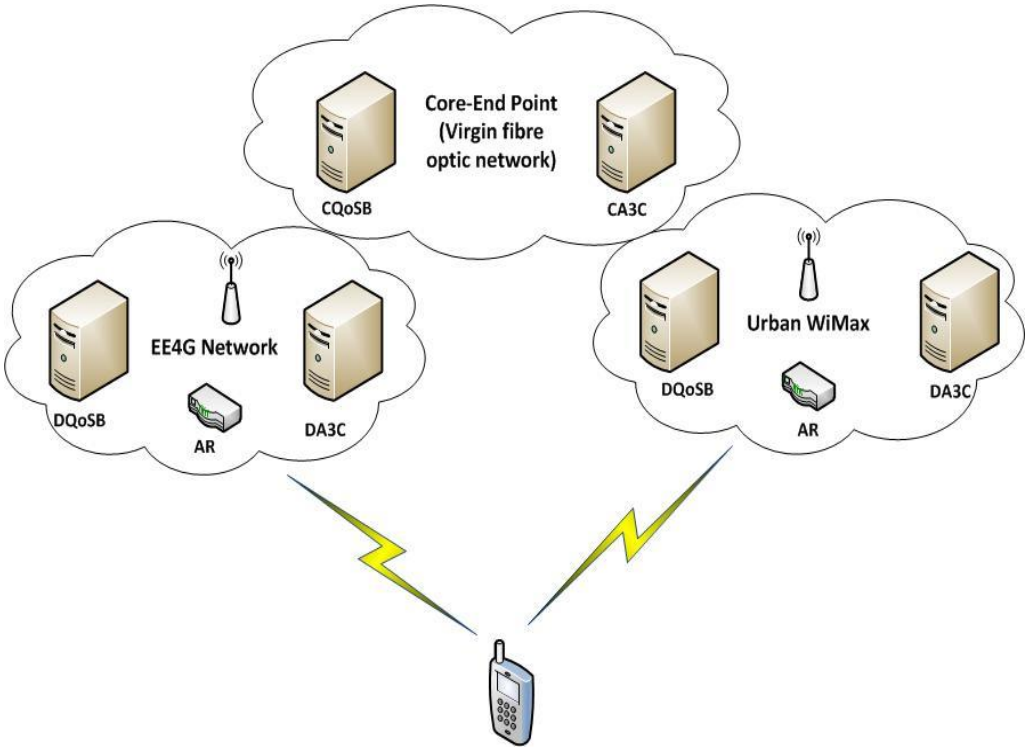

Figure 6-1. Y-Comm network structure (case study)

This case study assumes that the end user is an EE customer and that the EUD is already connected to the EE network. It then connects to the Urban WiMax network to get a high-speed connection. Because Urban WiMax provides high-speed connections in specific areas, this vertical handover occurs automatically, so no effort is required from the EUD. This is a fundamental feature of the Y-Comm network design.

When the EUD starts to connect to Urban WiMax (vertical handover from the EE), the EUD sends a combined message to the destination network (Urban WiMax). The Urban network authenticator, which is responsible for authenticating new EUDs, responds by sending an authentication request message to the EUD. In order to receive an authentication key to access Urban WiMax, the EUD then sends a message to the EE network, with which it is already connected and authenticated. This message is passed

from the EE authenticator to the EE DA3C server and then to the VFO CA3C server. The CA3C examines the service level agreement for the user's access to the network. If the EUD is allowed to access the Urban network, the CA3C will send a message to the Urban WiMax DA3C sever to notify the destination network of this. A message is then passed to the Urban WiMax authenticator, which sends a message to the EUD, authorising it to use the network.

Figure 6-2 is the vertical handover sequence diagram for this case study. The mechanism illustrated was not designed by this research, but by [6]. It is mentioned here to explain the applicability of the proposed SMS4HN to the Y-Comm network environment.

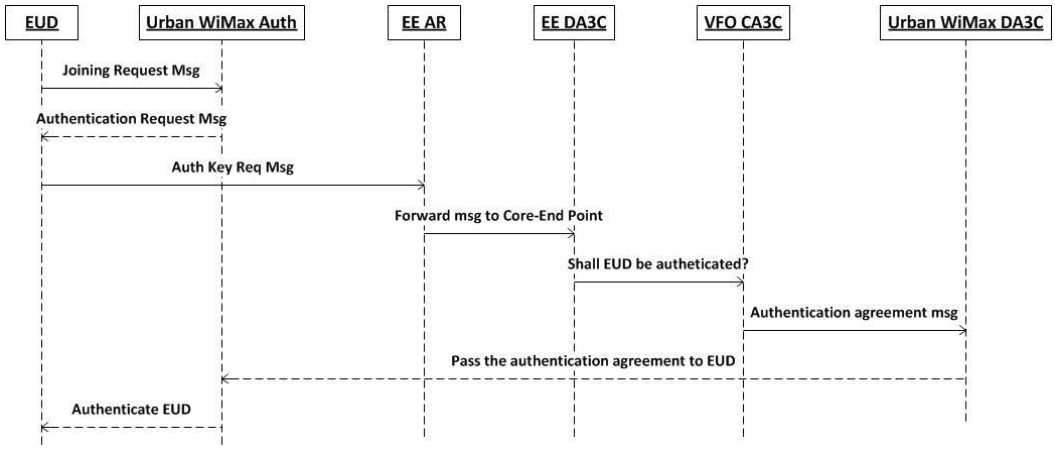

Figure 6-2. EUD vertical handover sequence diagram

All the Y-Comm network components, such as the CA3C and the AR, represent managed objects, and the proposed SMS4HN is able to interact with and enforce the appropriate policies on them. The next section explains the SMS4HN processes and how it interacts with these managed objects.

6.4 The SMS4HN in the Y-Comm network

This section illustrates the applicability of the SMS4HN to the Y-Comm network environment using a case study. As mentioned in Chapter 5, the proposed SMS4HN is a policy-based system, so a case study will be used to prove that the SMS4HN is able to enforce an appropriate policy in the security management system for 4G heterogeneous networks. This proposed policy-based system is implemented using Ponder2, and the self-managed cell is part of design solution for this Y-Comm network environment.

This case study assumes that the EUD has the privilege to access, modify, and manage network resources on the Urban network. This means that the EUD's identity is important, and that the theft of this identity might harm the network, as explained in Chapter 3.

In the case study, an attacker is trying access sensitive data on the GSM network [84]. The attacker can also do this in current Y-Comm networks by executing malware from an EUD to conduct malicious activities. The EUD may download this malware from an official application store or from the web. (The openness of Y-Comm networks makes this kind of attack more likely than on previous networks, as explained in Chapter 2.) The IA is responsible for detecting this malicious activity using the mechanism presented in Section 6.5. It then sends a report to the security engine including a mobile equipment identifier, the attack type, attack date, and attack time. The security engine follows the ITU-T recommendation, which states the need to isolate a dangerous EUD, and enforces the appropriate policy. In this situation, this is an obligation policy that

removes the user's access when it is triggered by a malicious event. To do this, the proposed system has two enforcement points, which are explored in the next subsection.

6.4.1 Policy enforcement points in the Y-Comm network

The two policy enforcement points in the Y-Comm network are in the Urban AR and the VFO CA3C server. The Urban AR is responsible for authenticating Urban network users and stops the EUD from gaining service connections. The security engine removes the user's access in the network-level agreement, which is located in the VFO CA3C. Figure 6-3 illustrates the two PEP locations.

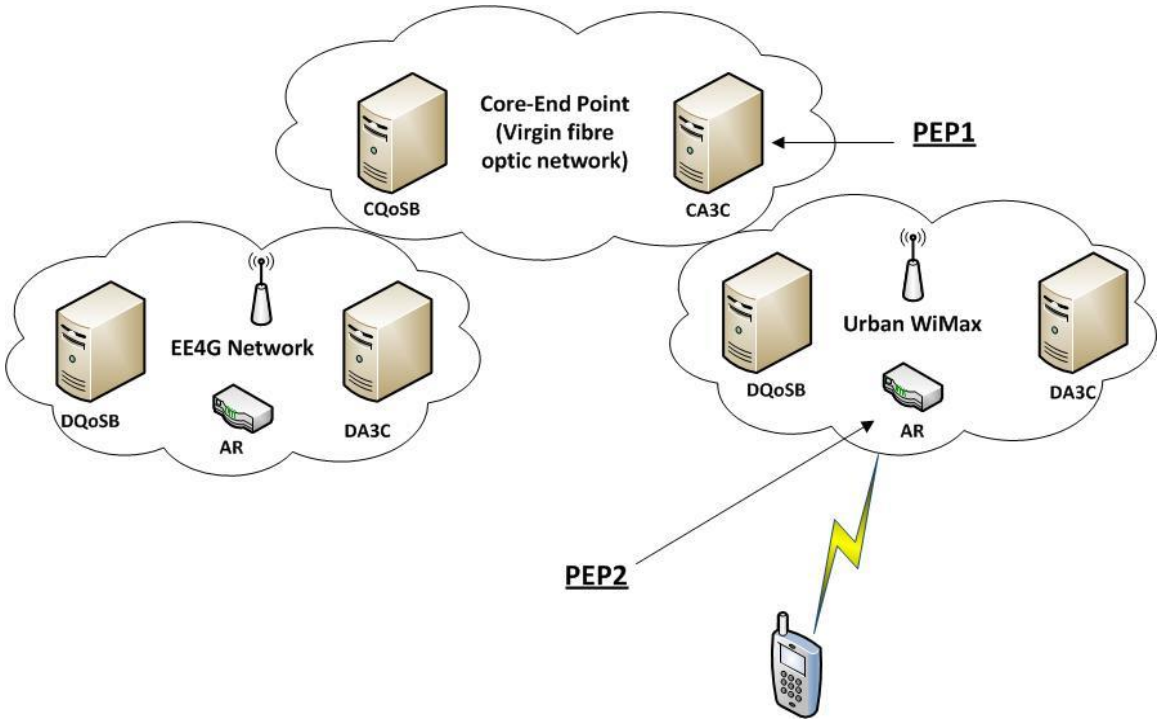

Figure 6-3. Location of the policy enforcement points in the Y-Comm network architecture

The SMS4HN enforces the CA3C, which stops the user's access to the network. However, when the NLA is modified, the EUD may still connect to a peripheral network, because it has already received authorisation. Therefore, the second enforcement point stops providing the EUD with connection services. Thus, the dangerous EUD is isolated. (This isolation was explained in Chapter 2 as a function set of the ITU-T M.3400 recommendation concerning dangers to the network.)

6.4.2 PAF in the Y-Comm network

The policy authorisation framework is part of the Ponder2 policy specification language. Chapter 3 mentions that the PAF in Ponder2 supports two PEPs to prevent the subject (the EUD) from harming the target (the Y-Comm network). The second point prevents the subject from receiving authorisation to access the target. Figure 6-4 illustrates the PAF for the SMS4HN in the Y-Comm network.

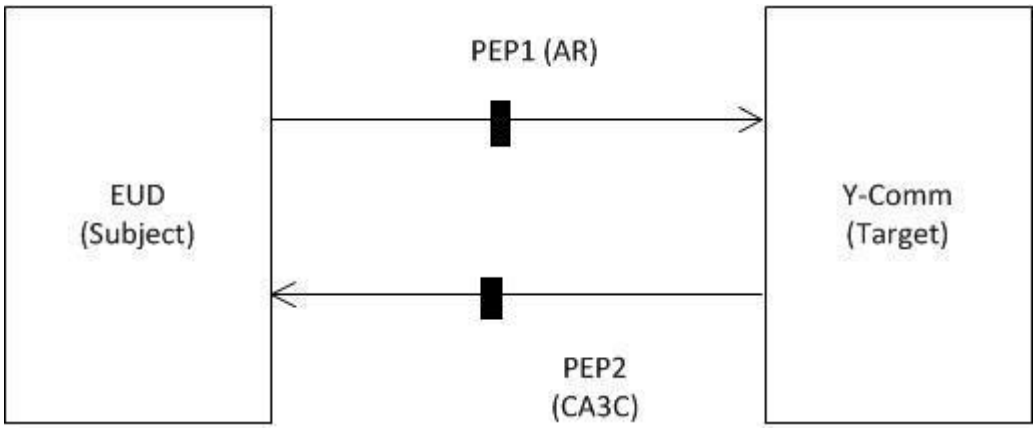

Figure 6-4. PAF of the Y-Comm network

6.4.3 The self-managed cell in the Y-Comm network environment

The SMC presents the interaction loop between the proposed policy-based system and the Y-Comm network components. The SMC concept has been designed by Ponder2 to support the dynamic policy-based system, as explained in Chapter 3. In this case study, the SMC of the proposed security management system works as a policy feedback loop. Figure 6-5 illustrates this loop and the interaction between the managed objects (EUD, EE AR, and VFO CA3C) and the proposed SMS4HN.

This process is called a loop because the interaction starts from a managed object (EUD), goes through the SMS4HN to handle the event, then returns to the managed object (EE AR, VFO CA3C). The EUD triggers the SMS4HN when the IA detects a malicious event, then a message is transferred through the event bus to the security engine. The security engine handles this event according to its obligation policies, which state that if a specific event occurs, a specific action should be taken. In step 4, as Figure 6-5 illustrates, the appropriate authorisation policy is enforced on the managed objects represented in the SMS4HN. For example, the VFO CA3C holds the NLA, and the EE AR works as an EUD authenticator.

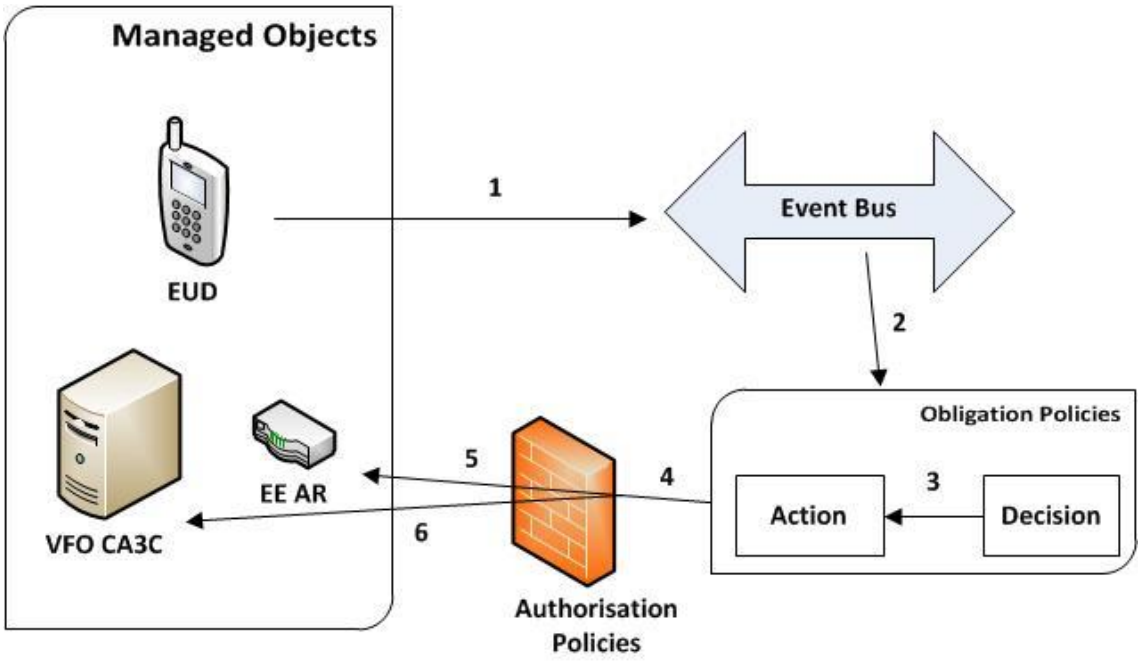

Figure 6-5. Policy feedback loop for the SMS4HN in the Y-Comm network

The complete scenario, from when the IA detects an attack to the last process that prevents the EUD from being used as an attack tool, is presented in the case study.

6.4.4 SMS4HN obligation policies

This subsection explains how this research project identified the SMS4HN obligation policies. Figure 6-6 shows a piece of code that explains how to create an ECA policy and how the SMS4HN interacts with the managed objects in the VFO (the Y-Comm case study). Figure 6-6 shows that the system creates an ECA policy to check whether there is a security violation detected by the IA and received as an event, then it receives the event, which contains the four attributes: EUD ID, attack type, attack date, and attack time. The SMS4HN interacts with the AR managed object to activate the stopAccess function, which means that it stops providing service to this EUD.

```
// Create a new policy to prevent the EUD from accessing the VFO network in case of attack
// Obligation policy for the case study of Virgin Fiber Optic with EE4G and Urban WiMax
        policy := root/factory/ecapolicy create.

        policy event: root/event/eudValue.
        policy condition: [ :eudID :AttackType :Date :Time |
                root print: "Checking End User Device: "+ eudID + " with attack type: " +
        AttackType.
                        AttackType == "ConfigAccess")].
        policy action: [
                root/DB keepRecord: eudID, AttackType, Date, Time.
                root/NLA removeAccess: eudID .
                root/AccessRouter stopAccess: eudID.
                root/warning setWarning: true; show ].
            root/policy at: "ConfigAccess" put: policy.
            policy active: true.
```

Figure 6-6. ECA policy in SMS4HN in the VFO network.

Another piece of code is shown in Figure 6-7, which depicts the creation of an event that is received from the EUD-managed object in the VFO network (Y-Comm case study). The system loads the event in the event bus, which transmits the events between the managed objects.

```
// Create a malicious event type for the SMS4HN in Y-Comm case study

        template := root/factory/event create: #( "eudID" "attackType" "attackDate"
        "attackTime" ).
        root/event at: "maliciousEvent" put: template.
```

Figure 6-7. Event template of SMS4HN in the VFO network.

6.4.5 Authorisation policies in SMS4HN

This subsection explores how the proposed SMS4HN uses authorisation policies in the VFO network. Figure 6-8 lists the commands required to specify a negative authorisation policy in the proposed SMS4HN in the VFO network.

```
// Authorisation policy for the VFO (The case study of Y-Comm)
newauthpol := root/factory/VFO/authpolicy.
root/VFO/authdom at: "authEUD" put:
                                    ( newauthpol
                                    subject: root/VFO/Urban/EUD
                                    action: "tcp"
                                    target: root/VFO/Urban/AR
                                    focus: "t").
root/VFO/authdom/authEUD reqneg.
root/VFO/authdom/authEUD repneg.
root/VFO/authdom/authEUD active: true.
```

Figure 6-8. Authorisation policy commands in SMS4HN.

The commands in Figure 6-8 first create an authorisation policy, then specify the subject, which is the EUD, and the target, which is the AR.

6.4.6 Case study results

This section presents results of an experiment within the case study. The last two sections explained the obligation policies and authorisation policies for the Y-Comm in the case of the VFO network attached to the EE4G and Urban WiMax networks. In this experiment, we created an event generator to simulate the EUD by generating malicious events to test the security engine's response in enforcing the authorisation policies. The SMS4HN interoperates these managed objects, which are components of the peripheral

and core networks (VFO, EE4G, and Urban WiMax). Figure 6-9 is a snapshot of the system as it responds to a malicious event in the case study.

Figure 6-9. Snapshot of SMS4HN after it detects a malicious event.

Figure 6-9 shows that the system's main steps after it detects a malicious event are to activate the policy to deny the EUD's access to the peripheral networks (EE4G or Urban WiMax), then to deny its access to the core network (VFO) and to keep a detailed record of the event.

This experimental system then created an output file to store the events that had occurred in the Y-Comm network. Figure 6-10 shows the output file of the malicious event record.

out.txt

EUD ID	AttackType	Date	Time	Peripheral Network	Core Network
458X87T88	ConfigAcs	07/02/2015	12:24	EE4G	VFO
359R87S98	ConfigAcs	07/02/2013	13:45	EE4G	VFO
287Q9R54Y	ConfigAcs	07/02/2015	14:13	EE4G	VFO
E95T1X4W6	ConfigAcs	07/02/2015	14:16	Urban WiMax	VFO
W47Q5V97E	ConfigAcs	07/02/2015	14:56	EE4G	VFO
93Q1C4E4L	ConfigAcs	07/02/2015	15:11	Urban WiMax	VFO
87W4C27YU	ConfigAcs	08/02/2015	08:54	Urban WiMax	VFO
97GKI3213	ConfigAcs	08/02/2015	11:23	EE4G	VFO
96QAZ123D	ConfigAcs	08/02/2015	12:57	EE4G	VFO
634RTH1F4	ConfigAcs	08/02/2015	13:10	EE4G	VFO
9RTU31425	ConfigAcs	08/02/2015	13:44	Urban WiMax	VFO
364RVY125	ConfigAcs	09/02/2015	11:41	Urban WiMax	VFO
63RRTG21W	ConfigAcs	09/02/2015	11:47	Urban WiMax	VFO
63QA45F2K	ConfigAcs	09/02/2015	14:36	Urban WiMax	VFO
2DQ763XZ5	ConfigAcs	10/02/2015	07:25	Urban WiMax	VFO
QRPOF862F	ConfigAcs	10/02/2015	08:16	EE4G	VFO
98EIN8532	ConfigAcs	10/02/2015	09:31	EE4G	VFO
82DEQ42V3	ConfigAcs	10/02/2015	10:34	Urban WiMax	VFO
1Q4G6E3R2	ConfigAcs	10/02/2015	11:26	EE4G	VFO
458X87T88	ConfigAcs	11/02/2015	12:24	Urban WiMax	VFO

Figure 6-10. Snapshot of the output file of the malicious events records.

6.5 Detection of a malicious event

6.5.1 Creating a normal model

This research has created a normal model of application behaviour, which enables the detection of new EUD activities. The IA compares any new EUD activity with the normal model and if it finds that the new activity is outside the model, it identifies this as malicious behaviour. There are three main steps in the creation of a normal model. First, the IA monitors the system calls of the Android OS applications. Next, it applies this monitoring task to many EUDs to collect as many application behaviours as possible. Finally, it uses a data mining technique to extract useful patterns from the collected information. These steps are explained in detail in the following subsections.

6.5.1.1 Monitoring the EUD system calls

The purpose of monitoring system calls is to have an overview of the behaviour of applications. The mechanism requires the monitoring of as many applications as possible, to maximise the accuracy of the normal model.

System calls in Android are application service requests from the operating system kernel. Android is an operating system based on the Linux kernel. There are 190 system calls in the Linux kernel, identified by unique numbers [76]. The analysis of these system calls for all applications provides a clear picture of the behaviour of the applications.

To monitor system calls in Android, it is important to understand the sequence of components that interact with the application and pass this system call to the kernel. First, the application makes a system call request, which is passed to the glibc library, comprising a set of functions in the Linux OS, such as *open()* and *read()*. After the request is passed to the glibc library, it is passed to the system call interface (SCI), which is responsible for executing system call requests in the kernel. Figure 6-11 shows the components between the applications and the kernel.

From this explanation, it is evident that the SCI passes all system call requests to the kernel. Thus, monitoring all the system calls that are passed through the SCI will show the exact behaviour of each application in the Android OS. All system call requests for each application are then stored in a file containing significant information, such as accessed data, opened files, and the number of system calls, which is sent to a server to create a normal model of the application's behaviour.

Monitoring the system calls in Android is similar to the approach used in [76, 99]. This case study provides a design solution based on this approach with some modifications and extensions.

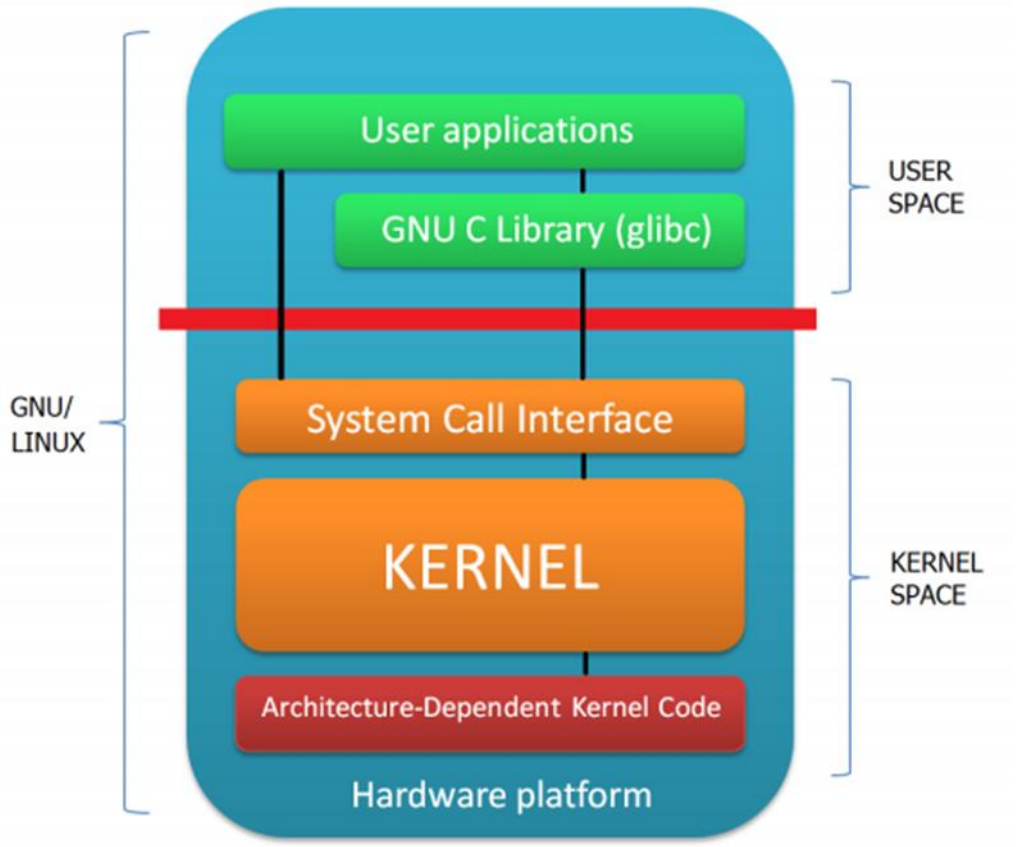

Figure 6-11. Linux user and kernel space [76]

6.5.1.2 Crowdsourcing

This IA detection mechanism monitors as many applications as possible in the Android OS, in order to make the normal model as accurate as possible. This can be done in cooperation with Google Inc. or by building a database for these behaviours. This research assumes that cooperation with Google Inc. is not possible, to prove that there

are alternative methods for creating a normal model. One of these involves building a database of normal application behaviours from scratch. It is difficult to complete this task using a single EUD, so the mechanism presented here uses many EUDs, in a process known as crowdsourcing.

In this context, crowdsourcing means collecting information from many Android devices. These EUDs contain software that monitors their normal applications and their system calls, then sends the information to a server. The server receives the large amount of information that it receives to create a normal model of application behaviour. This software has similar functions to that proposed by [76, 100].

6.5.1.3 Data mining

The last step of the IA detection mechanism is to extract useful patterns of data from a large database of application behaviours. As mentioned in the previous subsection, there will be large information files that contain details of Android application behaviour, so a method of extracting useful patterns from these large files is needed. Such a method is data mining, which uses statistics and the AI to find useful patterns in large datasets. The details of a similar approach can be found in [76].

There are many data mining techniques that can be used to extract useful information from a large collection of datasets. A survey of these techniques can be found in [101]. This case study suggests the use of *k-means* clustering, which divides large datasets into k groups or clusters, making it appropriate for the proposed IA. In this case study, $k = 2$ because two clusters (good and malicious) can explain the behaviour of the application.

Thus, the outcome of the above steps is a mechanism for the IA that recognises malicious behaviour. At the end of these three main steps, the IA will be able to compare any new activity with the normal model. Figure 6-12 shows the three main steps involved in creating a normal model.

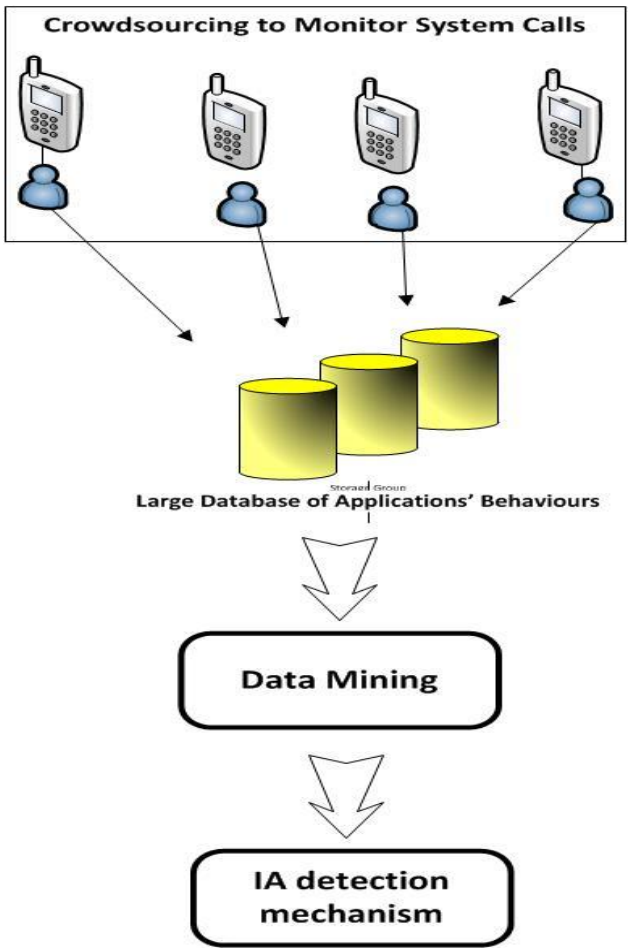

Figure 6-12. Main steps for creating a normal model of application behaviours

6.6 **Summary**

This chapter has presented a case study of Y-Comm with details of the mechanisms required to apply the proposed SMS4HN in the real word.

- It has explained vertical handover in the Y-Comm network using a case study, then given details of the proposed SMS4HN processes for the case study.
- It has explained the detection mechanism in detail. The steps followed by the IA include the creation of a normal model of application behaviours.

The next chapter evaluates the proposed SMS4HN and discusses the limitations of the research.

Chapter 7: Evaluation of the Security Management System for 4G Heterogeneous Networks

Objectives:

- Evaluate the proposed SMS4HN using several criteria;
- Discuss the limitations of the SMS4HN.

7.1 Introduction

So far, this thesis has set out the background, framework and implementation details of the security management system for the Y-Comm network. This chapter evaluates the proposed SMS4HN, in order to ensure that it will work in the Y-Comm network environment. A second goal is to compare the SMS4HN with related work. This chapter explains that the SMS4HN is a novel security management system for the Y-Comm network. The proposed SMS4HN is evaluated and compared with other security management systems, based on the following success criteria:

- The extensibility of the SMS4HN;
- The ability to modify the behaviour of the policy-based system while it is running;
- The self-management of the system;
- Its interoperability with the heterogeneous components of the Y-Comm network;
- Its scalability.

These criteria are used to compare the SMS4HN with the security management systems discussed in Chapter 2. The security management system in [78] is referred to as PWLSMS, and that in [79] as MIWSMS. The authors of the PWLSMS sought to build a system that would validate policy and generate new settings when the policy was violated in the network. They also built a local monitor for each area in the network, responsible for authenticating EUDs and enforcing policies. The MIWSMS differs in that it divides the network into policy zones. Its authors conclude that policy

enforcement is thus more efficient. Each policy zone includes a local server that is responsible for enforcing the policy. Chapter 2 explains their approaches and mechanisms in detail. This chapter evaluates the SMS4HN and compares it to the PWLSMS and the MIWSMS in terms of several criteria.

Section 7.2 evaluates the extensibility of the SMS4HN against these criteria. The SMS4HN's modifiability is discussed in Section 7.3, then Section 7.4 considers the self-management of the proposed system. Its interoperability with the Y-Comm network components is discussed in Section 7.5, while Section 7.6 discusses its scalability.

7.2 Extensibility

An important aspect of any security management system is its extensibility. Such a system can be extended in three ways: responding to additional malicious events, enforcing policies without additional hardware, and using additional communication protocols [45]. This section evaluates the extensibility of the SMS4HN in comparison to other security management systems. It was necessary to include this evaluation criterion in this project because the Y-Comm network is continuing to increase in size and because it enables new peripheral network providers to join the core network, as explained in Chapters 2 and 3.

7.2.1 The ability to respond to additional malicious events

The first dimension of extensibility to be addressed is the ability to respond to additional malicious events, which security management systems for heterogeneous

networks are expected to do. When an effective security management system detects an EUD performing functions that that it is not permitted to perform, this violation is called vertical privilege escalation. To prevent this type of violation, which is likely to harm the network resources and associated data, the security management system is expected to enforce a negative authorisation policy on the EUD.

The MIWSMS shows limited extensibility in this scenario. There are two procedures required to enforce a policy on an EUD that violates the access control rules and tries to escalate its privileges in a network. The MIWSMS specifies PRi <rj, Zi, Objk, T, d>, a policy rule which states that rj (the policy rule) has specified that the EUD has been denied access to Objk (for example, the AR) in zone Zi (the policy zone, for example, in the EE's peripheral network) during time interval T (for example, from 12/8/2014 to 30/11/2014). The implementation rule for this policy is the following: IRx < Ui, rj, Servk, Zi, T, d, neti>. Under this rule, the user Ui (EUD) is denied access to the network service Servk (for example, the connection service of Y-Comm) in zone Zi during time T, and neti (for example, the CA3C in Y-Comm). Due to its large scale and the necessity of sharing connection services between peripheral networks, it is difficult to divide the Y-Comm network into policy zones. Therefore, these procedures are not suitable for the Y-Comm network, because they require the identification of the EUD zone. These rules are also difficult to implement, making them inefficient for large-scale networks.

The SMS4HN responds to vertical privilege escalation with a specific policy that controls the managed objects. It specifies one obligation policy, and when the condition

in this policy is met, the negative authorisation policy is enforced on the EUD to prevent vertical privilege escalation. The following code snippet creates a template for the privilege escalation event in the SMS4HN. It also loads the EUD managed object, giving it a vertical privilege escalation type. This piece of code is reproduced here to explain that the SMS4HN is able to define the vertical privilege escalation event and respond to this event.

```
template := root/factory/event create.

root/event at: " privilegeEscale" put: template.

root/factory at: “EUD” put:(root load: “EUD”).

root at: “EUD” put:(root/factory/EUD event: root/event/privilegeEscale).
```

Figure 7-1. The commands used to create a template for the vertical privilege escalation of an event.

The piece of code in Figure 7-1 can be dynamically loaded into the SMS4HN, and after the new event type is loaded into the system, it is able to target and respond to this event.

The following code snippet specifies how the SMS4HN responds to the privilege escalation event.

```
// create a new policy to prevent the vertical privilege escalation (vpe)
policy := root/factory/vpepolicy create.

policy event: root/event/eudValue.

policy condition: [ :eudID :AttackType :Date :Time |

        root print: "Checking End User Device: "+ eudID + " with attack type: " +
        AttackType.

         AttackType == "VerticalEscale"].
         policy action: [
                root/DB keepRecord: eudID, AttackType, Date, Time.   ].
            root/policy at: "VerticalEscale" put: policy.
            policy active: true.

negativeAuth := root/factory/authpolicy.
root/authdom at: “authEscale” put:
                                    (negativeAuth
                                    subject: root/domainA/EUD
                                    action: “tcp”
                                    target: root/domain/AR
                                    focus: “t”).
root/authdom/authEscal reqneg.
root/authdom/authEscal active: true.
```

Figure 7-2. The commands used to specify the negative authorisation policy for an EUD.

Based on the above piece of code, the SMS4HN loads the event, checks whether it meets the policy condition, then performs the policy action, enforcing a negative authorisation policy on the EUD that has performed a vertical privilege escalation.

These examples show that the SMS4HN has better extensibility than other security management systems. This superior extensibility is the result of two factors: it is easy to control the Y-Comm network components because they are already designed as managed objects, and the SMS4HN has the self-management ability to handle an event and enforce a policy on managed objects.

The SMS4HN responds to a vertical privilege escalation in line with the ITU-T Recommendation M.3400, which has a function set that administers the external access control function set. This function set supports the security management system, allowing it to control what the user can do to any given resource and validating the user's permissions and credentials [102].

7.2.2 The ability to enforce policies without additional hardware

The second facet of extensibility is the ability to enforce policies without additional hardware. The PWLSMS is a security management system for both wired and wireless networks, whose designers sought to protect these networks from misuse by their users. According to them, this policy-based system is extensible because it adapts to new types of network equipment. The key architecture design is a local monitor, designed to sniff the data traffic and detect malicious behaviour. However, it should be asked how the system builds local monitors for a wide area network. Building many local monitors is

costly in time and resources. The added financial cost of providing such a service increases costs for end users in turn, but this violates the 4G requirement, discussed in Chapter 2, for a low-cost service.

The proposed SMS4HN, by contrast, achieves its security management goals without incorporating additional hardware. It interacts with the Y-Comm network components and sends messages to these existing components. The ability to achieve the security management system goals in a Y-Comm network without additional hardware is a strength of the SMS4HN.

7.2.3 The ability to communicate among managed objects using additional communication protocols

The third aspect of extensibility is the ability to communicate among managed objects using additional communication protocols. The policy-based system in a dynamic environment must be extensible in that the many communication protocols which support various technologies are themselves supported. This is important because different network technologies may require different protocols, such as IEEE 802.11, for the components to communicate. The local monitor in the PWLSMS communicates with the policy engine through XML messages [78]. However, there is no mention of whether the system supports other communication protocols.

Unlike the PWLSMS, the SMS4HN is able to use additional communication protocols, meaning that it has better extensibility than the PWLSMS.

In conclusion, the SMS4HN is able to respond to additional malicious events, whereas the MIWSMS has limited extensibility in its response to additional malicious events. In addition, the SMS4HN outperforms the PWLSMS in its ability to enforce policies without additional hardware and to use multiple communication protocols.

7.3 Modifiability

An important aspect of security management systems for mobile networks is modifiability, meaning the ability to modify the behaviour of a policy-based system during its run time without affecting other network services. For example, there should be no need to restart or recompile the system. This section evaluates the modifiability of the SMS4HN in comparison with the PWLSMS [78], assessing their respective ability to modify their behaviour without the need to restart or recompile the system. This research uses modifiability as an evaluation and comparison criterion because the Y-Comm network environment is expected to provide connection services at any time without affecting other services, which is the one of the 4G requirements explained in Chapter 2.

The PWLSMS uses XSB Prolog to implement its policy-based system, and the system must be restarted in order to activate or deactivate a policy. Conversely, policies implemented in Ponder2 can be added and removed without editing the component codes and specifications. Figure 7.3 provides an example of how the SMS4HN modifies its behaviour.

```
root/policy at: "ConfigAccess" put: policy.

policy active: true.

root/authdom/authEUD reqneg.
root/authdom/authEUD repneg.
root/authdom/authEUD active: true.
```

Figure 7-3. Activation and deactivation policies in SMS4HN.

Based on the above piece of code, each policy can be activated or deactivated easily with a one-line command, even after the policy has been specified. Therefore, a modification is achieved in SMS4HN without requiring the system to stop or recompile.

7.4 **Self-management**

Reducing human effort requirements for managing networks is an important consideration in policy-based systems, because they are expected to be self-managed and should respond instantly to malicious events without waiting for human input. This section evaluates the SMS4HN and compares it to the PWLSMS in terms of self-management ability. This was considered an evaluation criterion because Y-Comm is a large-scale network needing a policy-based system with low human effort requirements, thus reducing the cost of managing the network, as explained in Chapter 2.

The PWLSMS separates policy specification from policy management, which makes the policies robust. Although this feature is required for policy-based systems to

accommodate changes in network topology, this mechanism has a major drawback with regard to the detection of malicious activity by the policy engine. The local monitor in the PWLSMS's proposed architecture is responsible for monitoring all activities in the host and network equipment, but it does not monitor EUD activities. Therefore, when the policy engine is responsible for detection, the efficiency of malicious behaviour detection is reduced, because such behaviour can occur in different parts of the network, including in the EUD. In addition, the mechanism that detects malicious behaviour uses data traffic analysis, which may be misled or fail to detect real malicious behaviour.

The proposed SMS4HN improves the efficiency of detection by separating responsibility for it from the security engine. The IA detection mechanism depends on a deeper analysis of the malicious activities that occur in EUDs. Thus, malicious behaviour can be detected and reported to the security engine instantly, enabling it to enforce the appropriate policy. In addition, this research implements the SMS4HN as an SMC, as explained in Chapter 5. The SMC provides a loop of interaction that starts at the EUD, moves through the SMS4HN, which responds to this event, and ends with the enforcement of policies using Y-Comm network components, such as the AR and CA3C.

The use of managed objects in the SMS4HN also has several advantages. First, the system is able to manipulate them in the same way for management purposes. Second, when managed objects are specifically designed for Y-Comm network components, this increases their reusability for other purposes in the security management system. Third,

managed objects are transparent in the SMS4HN, which enables them to be controlled, regardless of their low-level specification details.

The PWLSMS uses XML to configure the policy system for a wireless networks' security management system, which has two major drawbacks. First, XML is hard to debug and read. Second, large pieces of code are needed to decode the XML messages received by a managed object [10].

The proposed SMS4HN uses PonderTalk to control the managed objects, as explained in Chapter 5. PonderTalk has been proposed by [10, 103] to overcome the drawbacks associated with XML and to provide a high-level language to control and configure the Ponder2 system. It allows messages to be sent to managed objects so that they can be controlled. PonderTalk ensures that the SMS4HN does not need to know the low-level details of various devices. This ability makes the SMS4HN suitable for Y-Comm network environments that contain a range of small devices and different network technologies.

7.5 **Interoperability**

Another important factor which this research considers is the SMS4HN's ability to work with the heterogeneous components of the Y-Comm network. It is important to control and send messages to the components, enabling the secure management of the Y-Comm network. Interoperability has been considered in this section because the Y-Comm network contains several heterogeneous components. Therefore, the Y-Comm network requires a security management system that is able to work and enforce

policies in a heterogeneous environment. This section evaluates the SMS4HN and compares it to the PWLSMS's ability to work with heterogeneous components.

The SMS4HN is designed to work with heterogeneous components as managed objects, which are wrapped in a piece of code that is used to interpret the messages they receive from other managed objects. These messages hold task commands, such as removing a user's access. Thus, interoperability between the managed objects that are components of the Y-Comm network and the SMS4HN is improved.

The PWLSMS uses XSB Prolog to program its policy engine. Its designers argue that Prolog supports well founded semantics. However, it has several drawbacks. For example, some Prolog compilers do not support the modules [104]. In addition, portability across real-world systems is a problem in Prolog [105]. Finally, [106] evaluated the performance of Prolog and found it to have more performance issues than other programming languages.

Based on the above discussion, the SMS4HN is seen to have better interoperability than the PWLSMS.

7.6 **Scalability**

An final important attribute of policy-based systems for large-scale networks is their scalability. According to Strassner [45], there are three types of scalability in policy-based systems: "(1) the ability to add capacity to existing resources, such as adding memory"; "(2) the ability to add additional network components, such as servers," and

in our case, the ability to add a new peripheral network to the Y-Comm network; and "(3) the ability to add logical processing power." In this research, scalability is taken as the ability of the proposed system to add a new peripheral network and new EUDs. Scalability was considered in this evaluation because the Y-Comm network has a constantly growing number of EUDs joining the network. This section evaluates the proposed SMS4HN's scalability and compares it with that of the MIWSMS [79].

A significant scalability limitation of the MIWSMS is that the crucial design requires a Policy Zone Controller located in the Local Role Server. The MIWSMS architecture divides the network into zones to enforce its policies. This design has a high cost and is inefficient for large-scale networks, requiring many servers and additional network resources. In addition, the MIWSMS is impossible to extend to and work on the Y-Comm network. The local monitors are also connected to WDPMans, which is a module that holds the policy engine via a wired interface, because it is more secure against DoS attacks, but this does not support the scalability of the policy-based system, because large-scale wireless networks use additional secure protocols to communicate between the components.

The SMS4HN uses an event bus, which allows it to separate the services into the event's sender and recipient. Thus, new event creators do not disturb the behaviour of the SMS4HN. For example, when new EUDs join the Y-Comm network, which allows these EUDs to send events to the security engine, the engine is able to receive these events without stopping the system. Therefore, using the event bus in the SMS4HN increases the successful scalability of the proposed system.

7.7 Limitations of the SMS4HN

This section discusses the function-related limitations of the SMS4HN as presented. No evaluation of the SMS4HN's performance was conducted, because it was not integrated into the Y-Comm network. When integration is completed, accurate performance measurement will be possible, and the SMS4HN's response time to malicious events can be measured more accurately. In addition, once integration is completed, the wrapping codes for the AR, CA3C, and NLA managed objects responsible for interpreting PonderTalk messages will be added to the Y-Comm.

A recovery method for an isolated EUD to use the Y-Comm network after isolation was not proposed in this framework. The solution to this recovery problem is proposed for future work.

The SMS4HN is a large system that requires significant effort for its integration into the Y-Comm network. Therefore, additional research on the Y-Comm network architecture is required to deploy and apply it in the real world, which constitutes a limitation of the present study. Future work in cooperation with Y-Comm network research studies is proposed to overcome the limitations of the SMS4HN.

7.8 Summary

This chapter has discussed and evaluated the proposed SMS4HN using several criteria.

- First, the extensibility of the SMS4HN was compared to two other security management systems: the PWLSMS and the MIWSMS.
- Next, this chapter evaluated the modifiability of the proposed policy-based system.
- The self-management of the SMS4HN was then discussed and compared with other security management systems.
- There was then an evaluation of the interoperability of the SMS4HN's components in the Y-Comm network.
- The scalability of the SMS4HN was discussed next.
- The final section addressed some limitations of the system.

The next chapter concludes this research study and presents plans for future research work on the proposed security management system.

Chapter 8: **Conclusion and Future Work**

Objectives:

- Present a summary of the thesis;
- Present the main contributions of the thesis;
- Describe intended future work on this research project.

8.1 Summary of the thesis

This thesis began by presenting the background to the study, including the evolution of mobile networks, then discussed the challenges to mobile computing, specifically a number of security issues. Next, it explained 4G mobile networks and heterogeneous networks, analysed the security of the Y-Comm network, and provided an overview of security policy types and access control models. Definitions of security management were given, then policy specification languages and their strengths and weaknesses were discussed. The choice of Ponder2 was explained and justified, and similar policy-based security management systems were discussed. Chapter 2 ended with an explanation of why this research study followed the ITU-T recommendations to build the proposed security management system.

Chapter 3 then explained the structure of future heterogeneous networks and provided a justification for the assumption of this research that theft of a user's identity is a dangerous violation that may lead to further attacks on the Y-Comm network itself. The unmet security requirements of 4G mobile networks were discussed next and the assumptions made in this research were stated. It was then important to present the fundamentals of Ponder2.

Chapter 4 established the framework for the new security management system and discussed details of its components. The framework is located on the top layer of the Y-Comm network and works as a management layer.

The specifications for the prototype implementation in the security engine were presented in Chapter 5. It described fundamental concepts, such the policy feedback

loop, which shows the interaction between the proposed system and the Y-Comm network's components, and the conflict resolution strategy, which is used for authorisation policy conflicts in the Y-Comm network. It was then explained how messages can be sent to managed objects using PonderTalk, and how the SMS4HN handles exceptions and errors using Ponder2 exception objects, such as *Ponder2ArgumentException*.

Chapter 6 reported a case study using the Y-Comm network that combined real, existing networks. This showed how an EUD moves from one peripheral network to another and discussed the role of the SMS4HN in the Y-Comm case study. The IA's detection mechanism, which is an anomaly-based approach that detects malicious activity in the Android operating system, was then explained.

Finally, Chapter 7 evaluated the proposed system and compared it to other policy-based security management systems against a set of criteria, chosen because they demonstrate that the SMS4HN is suitable for the Y-Comm network environment. These were extensibility, modifiability, interoperability, self-management, and scalability.

8.2 **Success criteria**

The thesis has answered the research questions presented in Chapter 1. It has presented a framework for the SMS4HN that works in the Y-Comm network's heterogeneous environment, enforces policies in that environment, and interoperates with its components. The proposed SMS4HN has been shown to fulfil the following success criteria:

- The SMS4HN interoperates with the Y-Comm network components, which are treated as managed objects. It sends messages to them using PonderTalk to enforce the appropriate policies when a malicious event occurs.
- The SMS4HN is extensible and is able to respond to additional malicious events and to enforce policies without the need for additional hardware, whereas other security management systems for wired and wireless networks need additional hardware components.
- The SMS4HN is self-managed and applies the policy feedback loop concept in the Y-Comm network. The loop is composed of two parts: obligation and authorisation policies.
- The modifiability of the presented policy-based system demonstrates its ability to activate and deactivate policies to change the behaviour of the system without requiring other system services to stop.

8.3 Contributions

The main contribution of this thesis is to propose and justify a novel security management system suitable for the Y-Comm network. The thesis has demonstrated its ability to work in the Y-Comm environment, as explained in Chapter 2. The contributions of this research study include:

- A new approach to protecting Y-Comm networks.

  The proposed approach prevented an EUD from being used as an attack tool in the Y-Comm environment. It followed ITU-T recommendation M.3400 to deal with the security violations in the network.

- **A policy-based framework for the Y-Comm network**. The purpose of this framework is to enforce policies in the Y-Comm network, a new architecture with heterogeneous components, which makes security management challenging. Thus, the network needs a novel framework that can respond to malicious events. The SMS4HN meets the security needs of the Y-Comm's design and is able to meet the criteria defined in Chapter 1.
- **A self-managed cell for the Y-Comm network to interact with managed objects**. The self-managed cell is represented as a policy feedback loop, which is triggered by the EUD and transmits an event to the security engine, then acts to enforce the appropriate policies.
- **A novel IA mechanism to detect malicious behaviour in an EUD**. The mechanism presented explained how to detect malicious behaviour in the EUD. The novelty of this mechanism is that the IA in the EUD interacts with the security engine in the CAS3C, thus providing better security for the Y-Comm environment.

8.4 Future work

The researcher aims to continue working on the SMS4HN so that it can be integrated into the Y-Comm network architecture in the future. Future work on the SMS4HN is expected to be conducted in cooperation with researchers from other institutions.

This future work includes the implementation of the IA in the EUD. The work will start using the Android operating system, and it will then move on to other common

smartphone operating systems. To make the system more efficient, the IA may generate a unique number for each EUD, which will be used as an identification number.

Another aim of future work is to write wrapping codes for components of the Y-Comm network, so they are able to interpret PonderTalk messages and complete tasks for security management purposes.

This research study aims in future to propose a mechanism that gives isolated EUDs their privileges back. This mechanism can be created to allow the SMS4HN to work in the Y-Comm network. Hence, this research will follow the ITU-T recommendations in the design of this mechanism.

This research study also aims to analyse the output file, which contains a record of the malicious events that have been detected. This analysis would be beneficial to update the security policies when new attacks occur in the Y-Comm network.

Finally, the study will propose a mechanism to evaluate the time it takes for the SMS4HN to respond to malicious events. This part of the evaluation will provide accurate measurements after it is integrated into the Y-Comm network architecture in the future.

References

[1] K. Kotapati, "Assessing security of mobile telecommunication networks," *Assessing Security of Mobile Telecommunication Networks,* vol. 1, pp. 20, 2008.

[2] N. Bikos and N. Sklavos, "LTE/SAE Security Issues on 4G Wireless Networks," *Security & Privacy, IEEE,* vol. 11, pp. 55-62, October 2012, .

[3] Y. Park and T. Park, "A survey of security threats on 4G networks," in *Globecom Workshops, 2007 IEEE,* Washington D.C, USA, 2007, pp. 1-6.

[4] M. Aiash, G. Mapp, A. Lasebae and R. Phan, "Providing security in 4G systems: Unveiling the challenges," in *Telecommunications (AICT), 2010 Sixth Advanced International Conference,* Barcelona, Spain, 2010, pp. 439-444.

[5] S. Hui and K. Yeung, "Challenges in the migration to 4G mobile systems," *Communications Magazine, IEEE,* vol. 41, pp. 54-59, Dec 2003, .

[6] N. Seddigh, B. Nandy, R. Makkar and J. Beaumont, "Security advances and challenges in 4G wireless networks," in *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference,* Ottawa, ON, Canada, 2010, pp. 62-71.

[7] Y. Zheng, D. He, W. Yu and X. Tang, "Trusted computing-based security architecture for 4G mobile networks," in *Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference,* Dalian, China, 2005, pp. 251-255.

[8] J. Oates, *Researching Information Systems and Computing.* Middlesborough, UK: Sage, 2005.

[9] P. D. Leedy and J. E. Ormrod, "Practical research," *Planning and Design,* vol. 8, pp. 20-43, Jan 2005, .

[10] K. Twidle, N. Dulay, E. Lupu and M. Sloman, "Ponder2: A policy system for autonomous pervasive environments," in *Autonomic and Autonomous Systems, 2009. ICAS'09. Fifth International Conference On*, Valencia, Spain, 2009, pp. 330-335.

[11] E. Lupu, N. Dulay, M. Sloman, J. Sventek, S. Heeps, S. Strowes, K. Twidle, S. Keoh and A. Schaeffer-Filho, "AMUSE: autonomic management of ubiquitous e-Health systems," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 277-295, 2008.

[12] A. Rahman and K. Sharma, "Fourth Generation of Mobile Communication Network: Evolution, Prospects, Objectives, Challenges and Security," *International Journal of Research in IT & Management*, vol. 2, pp. 2-13, February 2012, 2012.

[13] H. Chen, M. Guizani and W. Mohr, "Evolution toward 4G wireless networking [Guest Editorial]," *Network, IEEE*, vol. 21, pp. 4-5, February 2007, .

[14] S. Frattasi and A. Gimmler, "Potentials and limits of cooperation in wireless communications: Toward 4g wireless [guest editorial]," *Technology and Society Magazine, IEEE*, vol. 27, pp. 8-12, March 2008, .

[15] I. Recommendation, "Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000," International Telecommunication Union, Geneva, Tech. Rep. M.1645, January 2003.

[16] K. Kumar, J. Liu, Y. Lu and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications, Springer*, vol. 18, pp. 129-140, February 2013, .

[17] M. Satyanarayanan, "Mobile computing: the next decade," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 15, pp. 2-10, April 2011, .

[18] T. Hardjono and J. Seberry, "Information security issues in mobile computing," in *Eleventh International Information Processing Conference -Security'95*, Capetown, South Africa, 1995, pp. 143-151.

[19] M. La Polla, F. Martinelli and D. Sgandurra, "A Survey on Security for Mobile Devices," *Communications Surveys & Tutorials, IEEE*, vol. 15, pp. 446-471, March 2012, .

[20] T. Do and D. Gatica, "Where and what: Using smartphones to predict next locations and applications in daily life," *Pervasive and Mobile Computing*, vol. 12, pp. 79-91, June 2014, .

[21] N. Chilamkurti, S. Zeadally and H. Chaouchi, *Next-Generation Wireless Technologies: 4G and Beyond.* Germany: Springer, 2013.

[22] E. Dahlman, S. Parkvall and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband.* Waltham, USA: Academic Press, 2013.

[23] T. Miki, T. Ohya, H. Yoshino and N. Umeda, "The overview of the 4 th generation mobile communication system," in *Information, Communications and Signal Processing, 2005 Fifth International Conference,* Bangkok, 2005, pp. 1600-1604.

[24] E. Bou-Harb, M. Pourzandi, M. Debbabi and C. Assi, "A secure, efficient, and cost-effective distributed architecture for spam mitigation on LTE 4G mobile networks," *Security and Communication Networks*, vol. 6, pp. 1478-1489, Febryary 2012, .

[25] H. Gobjuka, "4G wireless networks: Opportunities and challenges," Cornell University Library, USA, Tech. Rep. VZ-TR-G1005309, 16 Jul 2009.

[26] M. Aiash, G. Mapp, A. Lasebae, M. Augusto, R. Vanni and E. Moreira, "A QoS framework for heterogeneous networking." in *Internation Conference on Wireless Networks 2011 (ICWN'11),* London, UK, 2011, pp. 142-158.

[27] F. Shaikh, "Intelligent Proactive Handover and QoS Management using TBVH in Heterogeneous Networks," 2010.

[28] F. Ghys and A. Vaaraniemi, "Component-based charging in a next-generation multimedia network," *Communications Magazine, IEEE*, vol. 41, pp. 99-102, Jan 2003, .

[29] M. Van Le, B. van Beijnum and G. Huitema, "A service component-based accounting and charging architecture to support interim mechanisms across multiple domains," in *Network Operations and Management Symposium. NOMS. IEEE/IFIP*, Seoul, South Korea, 2004, pp. 555-568.

[30] A. Guerrero-Ibáñez, J. Contreras-Castillo, A. Barba and A. Reyes, "A QoS-based dynamic pricing approach for services provisioning in heterogeneous wireless access networks," *Pervasive and Mobile Computing,* vol. 7, pp. 569-583, October 2011, .

[31] ITU-T, "X.805: Security architecture for systems providing end-to-end communications," International Telecommunication Union, Geneva, Switzerland, Tech. Rep. E24745, October 2003.

[32] R. Tafazolli, *Technologies for the Wireless Future: Wireless World Research Forum (WWRF)*. USA: Wiley. com, 2006.

[33] Q. Song and A. Jamalipour, "An adaptive quality-of-service network selection mechanism for heterogeneous mobile networks," *Wireless Communications and Mobile Computing*, vol. 5, pp. 697-708, September 2005, .

[34] G. Mapp, F. Shaikh, M. Aiash, R. Vanni, M. Augusto and E. Moreira, "Exploring efficient imperative handover mechanisms for heterogeneous wireless networks," in *Network-Based Information Systems, 2009. NBIS'09. International Conference On*, Indianapolis, IN, 2009, pp. 286-291.

[35] G. Mapp, D. Cottingham, F. Shaikh, P. Vidales, L. Patanapongpibul, J. Baliosian and J. Crowcroft, "An architectural framework for heterogeneous networking." in *Proceedings of the International Conference on Wireless Information Networks and Systems*, Colmar, France, 2006, pp. 122-134.

[36] G. Mapp, F. Shaikh, D. Cottingham, J. Crowcroft and J. Baliosian, "Y-comm: A global architecture for heterogeneous networking," in *Proceedings of the 3rd International Conference on Wireless Internet*, Brussels, Belgium, 2007, pp. 22-33.

[37] M. Aiash, "An integrated approach to QoS and security in future mobile networks using the Y-Comm framework," 2012.

[38] M. Augusto, R. Vanni, H. Guardia, M. Aiash, G. Mapp and E. Moreira, "MYHand: A novel architecture for improving handovers in NGNs," in *AICT 2013, the Ninth Advanced International Conference on Telecommunications*, Rome, Italy, 2013, pp. 211-218.

[39] G. Mapp, M. Aiash, A. Lasebae and R. Phan, "Security models for heterogeneous networking," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference*, Athens, Greece, 2010, pp. 1-4.

[40] G. Hoglund and J. Butler, *Rootkits: Subverting the Windows Kernel.* Japan: Addison-Wesley Professional, 2006.

[41] J. Bickford, R. O'Hare, A. Baliga, V. Ganapathy and L. Iftode, "Rootkits on smart phones: Attacks, implications and opportunities," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, Annapolis, MD, USA, 2010, pp. 49-54.

[42] (21st November 2011). *McAfee: Android is Sole Target of New Mobile Malware in Q3*. Available: http://www.networkworld.com/article/2183407/smartphones/mcafee--android-is-sole-target-of-new-mobile-malware-in-q3.html.

[43] R. Wies, "Policy definition and classification: Aspects, criteria and examples," in *Proceedings of the IFIP/IEEE International Workshop on Distributed Systems: Operation and Management*, London, UK, 1994, pp. 10-12.

[44] N. Damianou, A. Bandara, M. Sloman and E. Lupu, "A survey of policy specification approaches," *Department of Computing, Imperial College of Science Technology and Medicine, London,* vol. 3, pp. 142-156, April 2002, .

[45] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation.* San Franciso, USA: Morgan Kaufmann, 2003.

[46] M. Sloman, J. Lobo and E. C. Lupu, *Policies for Distributed Systems and Networks.* Bristol, UK: Springer, 2001.

[47] N. Damianou, "A Policy Framework for Management of Distributed Systems," 2002.

[48] P. Samarati and S. de Vimercati, "Access control: Policies, models, and mechanisms," in *Foundations of Security Analysis and Design*, 1st ed. ed., F. Riccardo and G. Roberto, Eds. Springer, 2001, pp. 137-196.

[49] J. Singh, J. Bacon and D. Eyers, "Policy enforcement within emerging distributed, event-based systems," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems,* Mumbai, India, 2014, pp. 246-255.

[50] H. Janicke, "The development of secure multi-agent systems," 2007.

[51] F. Cuppens, N. Cuppens-Boulahia and Y. Elrakaiby, "Formal specification and management of security policies with collective group obligations," *Journal of Computer Security,* vol. 21, pp. 149-190, February 2013, .

[52] N. Damianou, N. Dulay, E. Lupu and M. Sloman, "The ponder policy specification language," in *Policies for Distributed Systems and Networks,* 1st ed. ed., S. Morris, L. Emil and L. Jorge, Eds. Springer, 2001, pp. 18-38.

[53] M. Al-Sammarraie, "Policy-based approach for context-aware systems," 2011.

[54] V. Gligor, "Characteristics of role-based access control," in *Proceedings of the First ACM Workshop on Role-Based Access Control*, New York, NY, USA, 1996, pp. 10-23.

[55] R. Sandhu, E. Coyne, H. Feinstein and C. Youman, "Role-based access control models," *Computer*, vol. 29, pp. 38-47, February 1996, .

[56] A. Langsford, "OSI management model and standards," in *Network and Distributed Systems Management*, Boston, MA, USA, 1994, pp. 69-93.

[57] Z. Tu and Y. Yuan, "Critical success factors analysis on effective information security management: A literature review," in *Twentieth Americas Conference on Information Systems*, Savannah, USA, 2014, pp. 116-129.

[58] M. Sloman, J. Magee, K. Twidle and J. Kramer, "An architecture for managing distributed systems," in *Distributed Computing Systems, 1993., Proceedings of the Fourth Workshop on Future Trends Of*, Lisbon, 1993, pp. 40-46.

[59] B. Moore, E. Ellesson, J. Strassner and A. Westerinen, "Policy core information model–version 1 specification," RFC, USA, Tech. Rep. 3060, February 2001.

[60] B. Batista and M. Fernandez, "PonderFlow: A policy specification language for openflow networks," in *ICN 2014, the Thirteenth International Conference on Networks*, Nice, France, 2014, pp. 204-209.

[61] J. Lobo, R. Bhatia and S. Naqvi, "A policy description language," in *Proceedings of AAAI, Sixteenth National Conference on Artificial Intelligence*, New York, USA, 1999, pp. 291-298.

[62] L. Dhomeja, "Supporting policy-based contextual reconfiguration and adaptation in ubiquitous computing," 2011.

[63] W. Han and C. Lei, "A survey on policy languages in network and security management," *Computer Networks*, vol. 56, pp. 477-489, January 2012, .

[64] D. Lin, P. Rao, R. Ferrini, E. Bertino and J. Lobo, "A Similarity Measure for Comparing XACML Policies," *Knowledge and Data Engineering, IEEE Transactions On,* vol. 25, pp. 1946-1959, September 2013, .

[65] J. Bergstra and M. Burgess, *Handbook of Network and System Administration.* Norway: Access Online via Elsevier, 2011.

[66] C. Ramli, H. Nielson and F. Nielson, "The logic of XACML," in *8th International Symposium, FACS,* 1st ed. ed., A. Farhad and C. Peter, Eds. Elsevier, 2014, pp. 80-105.

[67] D. Ferraiolo and S. Gavrila, "Method and System for the Specification and Enforcement of Arbitrary Attribute-Based Access Control Policies," 12/366,855, 2009.

[68] J. Hoagland, "Specifying and Implementing Security Policies Using Lasco, the Language for Security Constraints on Objects," *arXiv Preprint Cs/0003066,* 2000.

[69] A. Alzahrani, "Efficient Enforcement of Security Policies in Distributed Systems," 2013.

[70] S. Saha and A. Nag, "Comparison of policy specification languages for access control," in *Proceedings of the CUBE International Information Technology Conference,* New York, NY, USA, 2012, pp. 764-770.

[71] J. Zhou, Q. Shen and Y. Xu, "Research and improvement of Ponder2 policy language," in *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference,* Zhangjiajie, China, 2012, pp. 455-458.

[72] E. Lupu, N. Dulay, M. Sloman, J. Sventek, S. Heeps, S. Strowes, K. Twidle, S. Keoh and A. Schaeffer-Filho, "AMUSE: autonomic management of ubiquitous e-Health systems," *Concurrency and Computation: Practice and Experience,* vol. 20, pp. 277-295, March 2008, .

[73] R. Neisse, P. Costa, M. Wegdam and M. van Sinderen, "An information model and architecture for context-aware management domains," in *Policies for Distributed*

Systems and Networks, 2008. POLICY 2008. IEEE Workshop, Palisades, NY, 2008, pp. 162-169.

[74] H. Zhao, J. Lobo and S. M. Bellovin, "An algebra for integration and analysis of ponder2 policies," in *Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop On*, Palisades, NY, 2008, pp. 74-77.

[75] E. Asmare and M. Sloman, "Self-management framework for unmanned autonomous vehicles," in *Inter-Domain Management*, 1st ed. ed., B. Arosha and B. Mark, Eds. London, UK: Springer, 2007, pp. 164-167.

[76] I. Burguera, U. Zurutuza and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for android," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, Chicago, IL, USA, 2011, pp. 15-26.

[77] J. Burns, A. Cheng, P. Gurung, S. Rajagopalan, P. Rao, D. Rosenbluth, A. V. Surendran and D. M. Martin Jr., "Automatic management of network security policy," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings*, Anaheim, CA, 2002, pp. 12-26.

[78] G. Lapiotis, B. Kim, S. Das and F. Anjum, "A policy-based approach to wireless LAN security management," in *Security and Privacy for Emerging Areas in Communication Networks, 2005. Workshop of the 1st International Conference*, Athens, Greece, 2006, pp. 181-189.

[79] S. Maity, P. Bera and S. K. Ghosh, "A mobile IP based WLAN security management framework with reconfigurable hardware acceleration," in *Proceedings of the 3rd International Conference on Security of Information and Networks*, New York, NY, USA, 2010, pp. 218-223.

[80] C. Perkins, P. Calhoun and J. Bharatia, "Internet engineering task force," Internet Engineering Task Force, Fremont, California , USA, Tech. Rep. RFC4721, January 2007.

[81] M. Aiash, G. Mapp, A. Lasebae, R. Phan and J. Loo, "Integrating mobility, quality-of-service and security in future mobile networks," in *Electrical Engineering and Intelligent Systems*, 1st ed. ed., L. G. Sio-Iong Ao, Ed. New York, USA: Springer, 2013, pp. 195-206.

[82] International Telecommunication Union (ITU-T), "Global information infrastructure, internet protocol aspects and next generation networks," ITU, Geneva, Tech. Rep. Y.140.1, November 2000.

[83] M. Aiash, G. Mapp, A. Lasebae, R. Phan and J. Loo, "A formally verified AKA protocol for vertical handover in heterogeneous environments using Casper/FDR," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, pp. 1-23, February 2012, .

[84] C. Guo, H. Wang and W. Zhu, "Smart-phone attacks and defenses," in *Third Workshop on Hot Topics in Networks,HotNets III*, San Diego, CA USA, 2004, pp. 124-137.

[85] S. Töyssy and M. Helenius, "About malicious software in smartphones," *Journal in Computer Virology*, vol. 2, pp. 109-119, August 2006, .

[86] M. Bishop, *Introduction to Computer Security*. India: Pearson Education, 2006.

[87] N. Damianou, N. Dulay, E. Lupu and M. Sloman, "A language for specifying security and management policies for distributed systems," *London: Department of Computing, Imperial College, Tech.Rep*, vol. 20, pp. 192-207, October 2000, .

[88] (25 November 2013). *Ponder2 wiki*. Available: http://www.ponder2.net/.

[89] A. Schaeffer-Filho, E. Lupu, N. Dulay, S. L. Keoh, K. Twidle, M. Sloman, S. Heeps, S. Strowes and J. Sventek, "Towards supporting interactions between self-managed cells," in *Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference On*, Cambridge, MA, 2007, pp. 224-236.

[90] S. L. Keoh, K. Twidle, N. Pryce, A. E. Schaeffer-Filho, E. Lupu, N. Dulay, M. Sloman, S. Heeps, S. Strowes and J. Sventek, "Policy-based management for body-sensor networks," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, Aachen, Germany, 2007, pp. 92-98.

[91] J. Chomicki, J. Lobo and S. Naqvi, "Conflict resolution using logic programming," *Knowledge and Data Engineering, IEEE Transactions On*, vol. 15, pp. 244-249, April 2003, .

[92] E. Lupu and M. Sloman, "Conflicts in policy-based distributed systems management," *Software Engineering, IEEE Transactions On*, vol. 25, pp. 852-869, October 1999, .

[93] N. Dunlop, J. Indulska and K. Raymond, "Methods for conflict resolution in policy-based management systems," in *Enterprise Distributed Object Computing Conference, 2003. Proceedings. Seventh IEEE International*, Brisbane, Qld., Australia, 2003, pp. 98-109.

[94] G. Russello, C. Dong and N. Dulay, "Authorisation and conflict resolution for hierarchical domains," in *Policies for Distributed Systems and Networks, 2007. POLICY'07. Eighth IEEE International Workshop On*, Bologna, 2007, pp. 201-210.

[95] 3GPP2, "3G mobile equipment identifier (MEID)," The Third Generation Partnership Project 2, France, Tech. Rep. S.R0048-A, August 2005.

[96] R. Oberg, *Mastering RMI: Developing Enterprise Applications in Java and EJB.* New York, USA: John Wiley & Sons, Inc., 2001.

[97] (28 August 2013). ***Android 'accounts for 79% of phone malware'***. Available: http://www.bbc.co.uk/news/technology-23863495.

[98] (13 May 2013). *Service speed*. Available: http://www.urbanwimax.co.uk/.

[99] T. Blasing, L. Batyuk, A. Schmidt, S. A. Camtepe and S. Albayrak, "An android application sandbox system for suspicious software detection," in *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference On,* Nancy, Lorraine, 2010, pp. 55-62.

[100] T. Buennemeyer, T. Nelson, L. Clagett, J. Dunning, R. Marchany and J. Tront, "Mobile device profiling and intrusion detection using smart batteries," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual,* Waikoloa, HI, 2008, pp. 296-296.

[101] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data,* 1st ed. ed., J. Kogan, C. Nicholas and M. Teboulle, Eds. Sunnyvale, CA, USA: Springer, 2006, pp. 25-71.

[102] ITU-T, "ITU-T recommendations, M.3400," International Telecommunication Union, Geneva, Tech. Rep. M.3400, March 2000.

[103] T. Kevin and L. Emil, "Ponder2 - the self managed cell," in *Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop On,* Palisades, NY, 2009, pp. 245-246.

[104] P. Moura, "Logtalk in Association of Logic Programming Newsletter," *ALP Newsletter,* vol. 17, pp. 12-21, August, 2004.

[105] J. Wielemaker and V. Costa, "Portability of prolog programs: Theory and case-studies," in *Joint Workshop on Implementation of Constraint Logic Programming Systems and Logic-Based Methods in Programming Environments (CICLOPS-WLPE 2010),* Edinburgh, Scotland, UK, 2010, pp. 13-25.

[106] L. Sterling, *The Practice of Prolog.* Melbourne, Australia: MIT press, 1990.

Appendix

In this appendix, some of the techniques necessary for the integration of the proposed SMS4HN into the Y-Comm network, such as sending messages to the managed objects using a web service, will be discussed. In addition, how other communication protocols are used to send messages to the managed objects in the SMS4HN will be explained. Then instructions on how to write and execute new policies in the SMS4HN will be provided along with an example of the process. Next, the technique for writing a managed object and the mapping from PonderTalk to the managed objects will be explained. Finally, instructions for the installation of Ponder2 and how to run the proposed SMS4HN will be provided.

A.1 Sending messages to managed objects using a web service

Chapter 5 mentioned that managed objects can communicate with each other in different ways and explained how they communicate using Java RMI. This section of the appendix shows how two managed objects (hardware or software) communicate using web service over the Y-Comm network. The following code example can be used to send an event from an EUD managed object to the security engine using a web service.

```
<wsdl:message name=”maliciousEvent”>
        <wsdl:part name=”port” type=”xsd:int”/>
        <wsdl:part name=”malaicousEvent” type=”xsd:string”/>
        <wsdl: part name=”args” type=”impl:ArrayOf_xsd_string/>
</wsdl>
```

This example shows the SMS4HN's ability to enable communication between managed objects using a different method from Java RMI, and it provides an example of the proper code for future enhancements to the SMS4HN.

A.2 Using other communication protocols with the SMS4HN

This research explained how the managed objects within the SMS4HN communicate using Java RMI. Section A.1 identified how messages can be sent to the managed objects using a web service. However, the Y-Comm network contains network technologies that may require other communication protocols. Thus, this section explains how to use a protocol that is not supported by ponder2comms.jar.

Ponder2 allows other protocols to be added to the SMS4HN, so the managed objects can communicate with each other. However, to add another protocol, a new protocol module should be written [88] and included as a jar file in the Ponder2 classpath. Once the module is added to the classpath, the module can be loaded when it is needed. The Ponder2 system does not require any other configurations or changes, which is an advantage of Ponder2.

A.3 Writing a new policy for the SMS4HN

In the future, the SMS4HN may require a new policy, so this section explains how to add a new policy for vertical privilege escalation, which occurs when an EUD performs an action it is not permitted to perform. As explained in Chapter 7, this action must be prevented. The policy states that whenever an EUD is detected performing actions that are not permitted, a negative authorisation policy should be enforced on the EUD. The steps to add a vertical privilege escalation policy are as follows:

- Create a file with a `.p2` extension in the top directory of the SMS4HN folder;
- Run the SMS4HN using an `ant` command;
- Executed the newly created file using `ant [the new created file].`
-

When the policy is executed and running, it has the ability to respond to the vertical privilege escalation. The next section explains in detail how to write the obligation policy for the vertical privilege escalation violation.

A.4 Writing a vertical privilege escalation policy

To write an obligation policy in Ponder2, an event template must be created and loaded into the event bus. When a predefined event occurs, the SMC triggers the obligation policy for that event. The specified policy checks the conditions and executes a predefined action. The event template for the vertical privilege escalation requires the following commands:

```
template := root/factory/event create.

root/event at: " privilegeEscale" put: template.

root/factory at: “EUD” put:(root load: “EUD”).

root at: “EUD” put:(root/factory/EUD event: root/event/privilegeEscale).
```

These commands are then loaded into the event template in the event bus, so when the vertical privilege escalation event is detected by the SMC, the policy for the predefined event is applied. A vertical privilege escalation policy should be created using the Ponder2 command:

```
policy := root/factory/vpepolicy create.
```

Then it should define the policy event using the following command:

```
policy event: root/event/eudValue.
```

The condition should be checked using the following Ponder2 command:

```
policy condition: [ AttackType == "VerticalEscale"].
```

It should then specify the obligation policy actions, which requires the following commands:

```
policy action: [

    root/DB keepRecord: eudID, AttackType, Date, Time.    ].

    root/policy at: "VerticalEscale" put: policy.

    policy active: true.

negativeAuth := root/factory/authpolicy.

root/authdom at: “authEscale” put:

                                    (negativeAuth

                                    subject: root/domainA/EUD

                                    action: “tcp”

                                    target: root/domain/AR

                                    focus: “t”).
```

Finally, a record of the event is saved in the database, and a negative authorisation policy is enforced on EUD that violated the vertical privilege escalation.

A.5 Writing a managed object

This section briefly explains how to create managed objects in the SMS4HN. The managed objects in Ponder2 receive and send PonderTalk messages, but the managed objects should be written in Java. A managed object in Ponder2 is annotated with PonderTalk bindings, such as `@Ponder2op()`, so it is able to receive PonderTalk

messages and perform the required actions. The `@Ponder2op()` annotation is located on the top of the method and contains the PonderTalk message name [88].

To write a managed object, the compiler must be told that it is a managed object. Do this using the following command:

```
class AccessRouter implements ManagedObject {
```

A constructor bound to PonderTalk is needed so the Ponder2 factors recognise the required call when it initiates the object. To do this, the following Ponder2op command is used:

```
//* Creates an instance of this object
  */
@Ponder2op("create")
public AccessRouter () {
}
```

In the previous command, the factory message *create* is used to create the object.

```
@Ponder2op("stopAccess:")
public AccessRouter (int eudID) {
}
```

In the previous command, the factory message *stopAccess* is used for a specific task in an access router managed object, and *stopAccess* receives the eudID (The EUD ID).

When PonderTalk commands are created for use in the SMS4HN, a Java method with the appropriate annotation and arguments is created. This step requires the following command:

```
@Ponder2op("PonderTalk_command")
public void method(args) {
}
```

`PonderTalk_command` is the name of the command PonderTalk sends. The number of arguments sent should match the arguments specified in the method.

A.6 PonderTalk method mapping

The mapping from PonderTalk commands to Java is explained in this section. There are three types of PonderTalk commands: unary, binary, and multiple arguments commands.

The following is a unary PonderTalk command example:

Java	PonderTalk

`@Ponder2op("setWarning")`
 	
`public void warnAdmin() {`
 	
`}`
	root/warning setWarning.

A binary command has only one argument, and an example of a binary command is as follows:

Java	PonderTalk

`@Ponder2op("removeAccess")`
 	
`public void removeEUDaccess(String eduID) {`
 	
`}`
	root/NLA removeAccess: eudID .

A multiple PonderTalk command contains multiple arguments separated by a ‘:’. The following is a mapping example:

Java	PonderTalk

`@Ponder2op("eudID:AttackType:Date:Time")`
 	
`public void keepRecordinDB( int eudID, String AttackType, String AttackDate, String AttackTime)`
	root/DB keepRecord: eudID, AttackType, Date, Time.

A.7 Ponder2 installation

This section explains how to install the Ponder2 system to run the SMS4HN. The only requirement for running the Ponder2 system is Java 1.5. The operating system used in this explanation is the Linux (Ubuntu) OS. Also, this research used Ponder2 version 2.3623. The steps to install Ponder2 are:

- Download the Ponder2 package from the following link: http://ponder2.net/cgi-bin/moin.cgi/Ponder2Downloads;
- Extract and unzip the files, and run the Ubuntu shell [103];
- Change the directory using the following command: `cd ponder2/p2src;`
- Enter the following command to install Ponder2: – ant install

The response from the Ubuntu system will be:

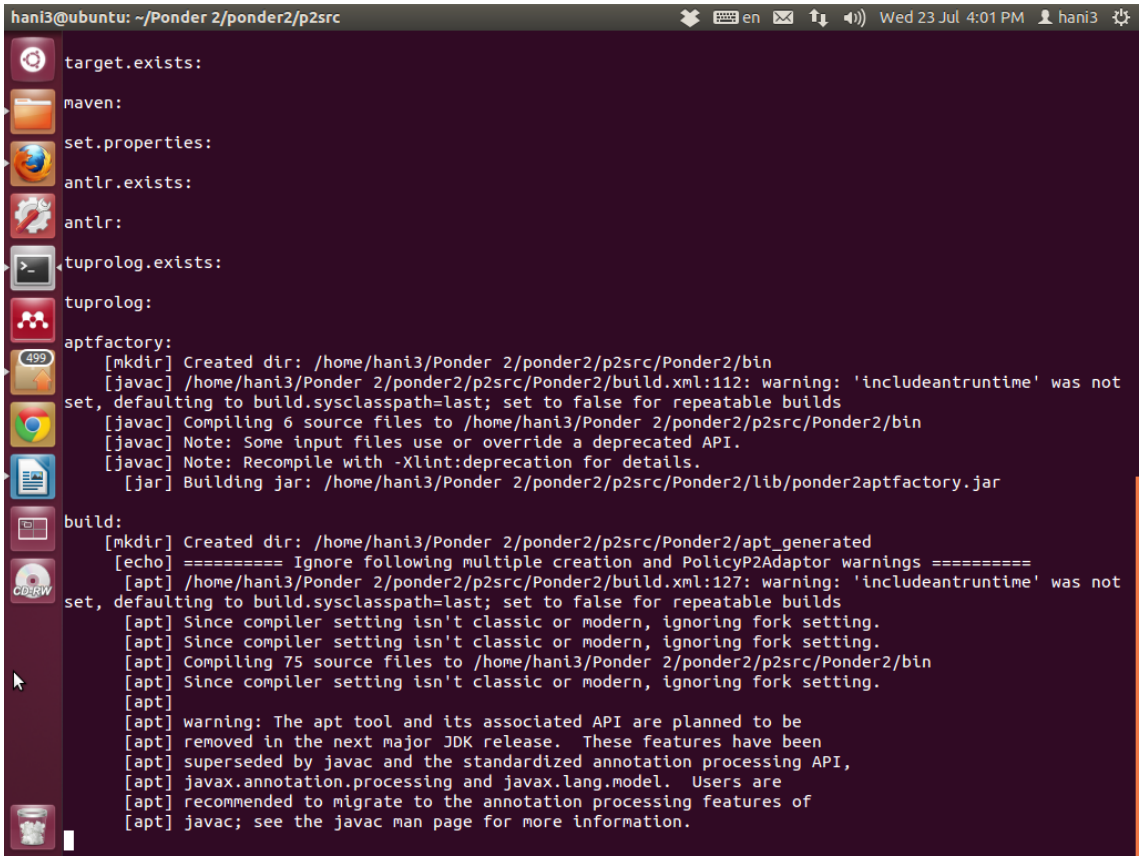

Figure A.8-1. Ubuntu response after Ponder2 system installation

A.8 Running the SMS4HN

To run the SMS4HN, Ponder2 must be installed. The steps to run the SMS4HN are as follows:

- Change the directory using the following command: `cd SecMangSys`;
- Enter the following command: `ant -Dboot=SMS5.p2`.

Ubuntu will respond as follows:

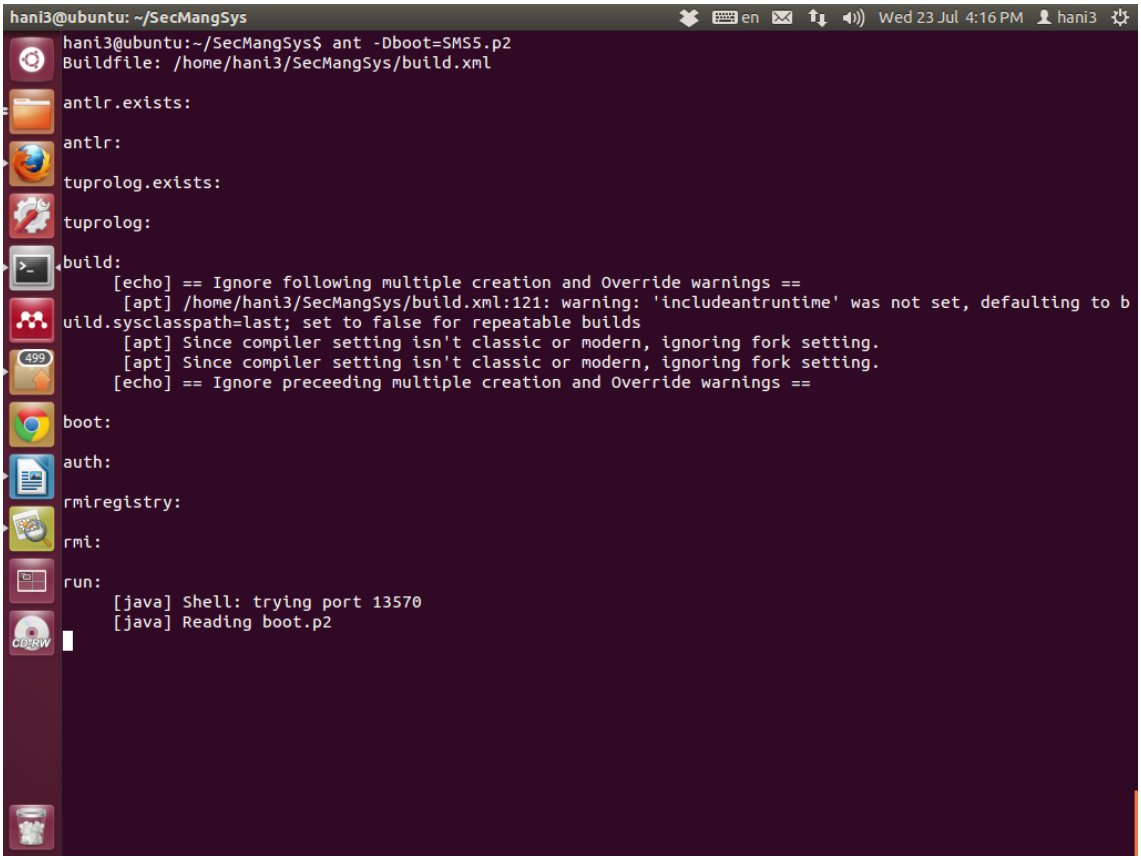

Figure A.8-2. Ubuntu response after running the SMS4HN